



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**OPTIMIZACIÓN DE PROCESOS PARA EL MANEJO DE CRÉDITOS Y
CUENTAS EN MICROFINANCIERAS**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERA EN COMPUTACIÓN**

P R E S E N T A N:

**JIMÉNEZ JIMÉNEZ YOLITZIN
SÁNCHEZ GARCÍA MONSERRAT**

**DIRECTOR DE TESIS :
ING. ORLANDO ZALDÍVAR ZAMORATEGUI**

MÉXICO D.F., MARZO DE 2016

Agradecimientos

Yolitzin Jiménez Jiménez

A mis padres, por enseñarme que los sueños se logran a base de esfuerzo y dedicación.

A mis hermanos, por darme la oportunidad de estar con ellos en nuestro crecimiento profesional y personal.

A mi amiga y compañera Monserrat Sánchez, por su cariño y paciencia, gracias por todos estos años de aventuras.

A la Facultad de Ingeniería por ser mi guía en la vida profesional. Gracias por esos momentos inolvidables.

A mi Universidad por enseñarme que todo es posible.

Al Ing. Orlando Zaldívar Zamorategui por su paciencia y guía, ya que sin su ayuda no habiéramos podido concluir. Gracias

Monserrat Sánchez García

A mis padres por su incondicional apoyo, guía, comprensión y por demostrarme que todo es posible.

A mis hermanos por acompañarme en este largo camino.

A la UNAM por permitirme vivir esta bella etapa de mi vida.

A la Facultad de Ingeniería y sus profesores que me formaron para alcanzar mis metas.

A Yolitzin por ser amiga y cómplice de tantos sueños.

Al ingeniero Orlando Zaldívar por guiarnos con paciencia a pesar de las adversidades.

¡Muchas gracias a todos!

ÍNDICE

Contenido	Pág.
Capítulo 1	- 1 -
Planteamiento del problema.....	- 1 -
1.2 Definición del problema	- 2 -
1.3 Objetivo general.....	- 5 -
1.4 Objetivo específicos	- 5 -
1.5 Alcance	- 5 -
Capítulo 2	- 6 -
Marco Teórico.....	- 6 -
2.1 Teoría relacionada con la problemática	- 7 -
2.1.1 Definición de microcrédito y microfinanzas	- 8 -
2.1.2 Antecedentes de las microfinancieras.....	- 9 -
2.1.3 Importancia de las microfinancieras	- 9 -
2.1.4 El financiamiento informal como alternativa.....	- 12 -
2.1.5 Sistema financiero mexicano	- 14 -
2.1.6 Características del sistema financiero mexicano	- 14 -
2.1.7 Descripción del sistema financiero mexicano	- 15 -
2.1.8 Plan de administración de una microfinanciera.....	- 18 -
2.1.9 Tecnologías microcrediticias	- 20 -
2.1.10 Ciclo de un crédito	- 21 -
2.2 Teoría relacionada con la ingeniería de software	- 26 -
2.2.1 Marco teórico. La ingeniería de software	- 26 -
2.2.2 Metodologías para desarrollo de software.....	- 28 -
2.3 Teoría de JAVA	- 37 -
2.3.1 Historia de JAVA	- 38 -
2.3.2 Características de la programación orientada a objetos	- 39 -
2.3.3 Características del lenguaje	- 43 -
2.4 Web Service	- 45 -
2.4.1 Concepto de Web Service	- 46 -
2.4.2 Estructura de Web Service	- 47 -
2.4.3 Arquitectura SOA.....	- 48 -
2.4.4 Protocolo SOAP	- 49 -
2.5 Lenguaje .NET.....	- 50 -
2.5.1 Características de .NET	- 50 -
2.5.2 Componentes del lenguaje .NET	- 51 -
2.5.3 Entorno integrado de desarrollo con Visual Studio.....	- 56 -

2.5.4 Entorno bancario y base de datos	- 57 -
2.5.5 Tecnología	- 58 -
2.5.6 Base de datos	- 58 -
Capítulo 3	- 63 -
Desarrollo del sistema de la Microfinanciera	- 63 -
3.1 Definición de requerimientos	- 64 -
3.1.1 Planificación del proyecto	- 66 -
3.1.2 Análisis de requisitos del software	- 69 -
3.2 Diseño del sistema y del software.....	- 69 -
3.2.1 Arquitectura del sistema.....	- 69 -
3.2.2 Estructura de datos	- 70 -
3.2.3 Detalle procedimental	- 73 -
3.2.4 Interfaces	- 76 -
3.3 Implementación y prueba de unidades.....	- 77 -
3.3.1 Implementación de la base de datos.....	- 78 -
3.3.2 Implementación de Web Service.....	- 79 -
3.3.3 Implementación de interfaz de usuarios	- 82 -
3.4 Integración y prueba del sistema	- 86 -
3.4.1 Algunas pruebas del sistema.....	- 90 -
3.5 Funcionamiento y mantenimiento.....	- 95 -
Capítulo 4	- 100 -
Resultados, impacto y conclusiones	- 100 -
4.1 Resultados.....	- 101 -
4.2 Impacto.....	- 115 -
4.3 Conclusiones	- 117 -
ANEXO A.....	- 119 -
ANEXO B.....	- 121 -
ANEXO C.....	- 122 -
ANEXO D.....	- 123 -
ANEXO E	- 124 -

Capítulo 1

Planteamiento del problema

1.1 Antecedentes

La situación social en la que nos encontramos en este siglo se ha marcado por una pobreza extrema en la sociedad mexicana; el INEGI en sus estadísticas menciona que la población en situación de pobreza es del 45.5%¹ del total de la población de la República Mexicana que representa 53,300,000 personas de 112,336,538 habitantes en México² de los cuales 8,851,080 viven en el Distrito Federal.

Las medidas que han surgido para ayudar a esta gran parte de la población son las sociedades financieras que a un menor costo otorgan préstamos en comparación a una institución financiera como un banco.

Actualmente una sociedad financiera se forma de manera inicial con pocos recursos, dependiendo de las personas en las cuales está enfocado, es decir, para ayuda de pequeños comercios, cultivo de tierras o para la compra de tractores.

Esta sociedad financiera, a la cual haremos referencia como la “*Microfinanciera*”, es una pequeña institución que proporciona préstamos a personas que cumplen con ciertos requisitos y que, debido a su crecimiento, se ve obligada a adquirir un nuevo sistema para tener un mayor control de los ingresos y egresos.

1.2 Definición del problema

En México, el sistema financiero es un ramo que está desarrollándose a pasos agigantados, gracias a lo cual ha habido un surgimiento de sociedades financieras con un gran mercado y que se fundan con la esperanza de un gran futuro.

Éste es el motivo que impulsa el planteamiento de esta tesis: atender a la demanda de un sistema capaz de soportar una gran cantidad de registros y brindar todos los requerimientos necesarios para poder manejar correctamente la información proporcionada por las personas que solicitan el servicio.

Actualmente, la *Microfinanciera* es una sociedad financiera popular encargada de realizar la captación de dinero de los clientes para poder llevar a cabo actividades de préstamo y ahorro y, de esta manera, realizar un flujo de dinero contribuyendo al crecimiento de los clientes y de la financiera misma.

¹Según informa el Consejo Nacional de Evaluación de la Política de Desarrollo Social (Coneval) en el anexo estadístico 2012 de la pobreza en México.

<http://www.coneval.gob.mx/Medicion/MP/Paginas/Pobreza-urbana-en-M%C3%A9xico-.aspx> , Enero 2016

² Página web: INEGI <http://cuentame.inegi.org.mx/poblacion/habitantes.aspx?tema=P>, Marzo 2015

La *Microfinanciera* cuenta con un sistema manual de captura de datos y con tres sucursales ubicadas en el centro del estado de Chiapas con las siguientes características:

- Los clientes se caracterizan por ser población de condición media
- Los créditos aperturados van de 100 a 120 mensuales
- El monto máximo de los mismos es un promedio de \$80,000
- El sistema de captura es manual, es decir, se requiere fotocopiar y capturar los datos en sistemas locales
- Cuenta con sistema de mensajería para que se pueda compartir la información en las diferentes sucursales

Debido al éxito obtenido, se analiza la posibilidad de un crecimiento de sucursales tanto en el mismo estado como en algunos otros, dado lo cual se considera al actual proceso de captura de datos y administración como ineficiente.

Actualmente, el proceso de alta de un crédito se lleva de la siguiente manera (Figura 1):

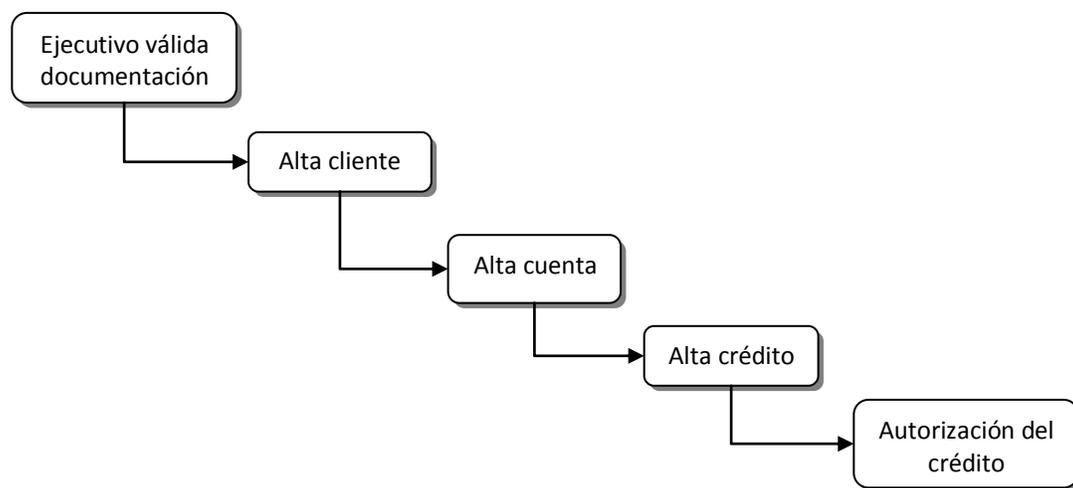


Figura 1. Alta de un crédito en la *Microfinanciera*

1. El cliente acude a la sucursal con un ejecutivo de la *Microfinanciera* y solicita un crédito. El ejecutivo es el encargado de solicitar la documentación necesaria para comprobar su identidad y capacidad crediticia (IFE, comprobante de domicilio, estado de cuenta, entre otros)
2. El cliente reúne los documentos requeridos; el ejecutivo valida que la documentación sea vigente y fidedigna
3. El ejecutivo da de alta la cuenta y cliente con los documentos proporcionados. En caso de faltar alguno se procede a solicitarlo nuevamente o a rechazar la solicitud

Los documentos proporcionados por los clientes son analizados, escaneados y capturados por los ejecutivos (Figura 2).

Errores frecuentes:

- El ejecutivo captura erróneamente algunos datos: RFC, CURP, etc.
- El ejecutivo captura erróneamente caracteres no válidos
- El ejecutivo captura datos no correctos
- El ejecutivo no captura datos obligatorios



Figura 2. Control actual del proceso de captura de datos

4. El ejecutivo da de alta el crédito. En su caso notifica al cliente que su crédito fue aprobado o bien pasa a gerencia para autorización
5. El gerente de la institución financiera es el encargado de la autorización del crédito

Errores frecuentes (Figura 3):

- La decisión de aceptar o rechazar créditos se deja a juicio del ejecutivo y/o analista de créditos. En ocasiones erróneamente se rechazan, lo que conlleva pérdida de tiempo, horas hombre y gastos administrativos
- Captura errónea de los datos necesarios para el proceso del crédito
- Dependencia de criterios personales de los empleados
- Rezago de validación de datos



Figura 3. Aceptación o rechazo de créditos a juicio de los ejecutivos

1.3 Objetivo general

La *Microfinanciera* busca la apertura de nuevas sucursales e implementar un sistema. Para ello se requiere garantizar la integridad de la información y la eficiencia en la apertura de clientes, cuentas y créditos.

- Garantizar la integridad de la información
- Mejorar el proceso de captura de la información de las clientes, cuentas y créditos
- Minimizar los errores de captura de la información
- Minimizar el tiempo de respuesta al cliente
- Garantizar la disponibilidad de la información
- Minimizar costos en los procesos de la financiera
- Evitar la decisión de autorización de créditos a juicio de ejecutivos
- Garantizar la adaptabilidad de los sistemas a nuevas aplicaciones
- Garantizar la escalabilidad del sistema para un futuro crecimiento

1.4 Objetivo específicos

La *Microfinanciera* busca garantizar los servicios con los que cuenta y, de esta manera, aumentar a corto plazo el número de clientes y sucursales.

Para lograrlo, se realizará un proceso de automatización del sistema de captura con lo que se garantizará un proceso eficiente.

1.5 Alcance

Este sistema de captura de clientes ayuda:

- Garantizar el menor tiempo para la apertura de los clientes
- Reducir costos de papelería
- Mejorar el rendimiento de los ejecutivos de cuentas
- Menor tiempo de capacitación a los miembros de ejecutivos de cuentas
- Mayores facilidades a los clientes para contar con un crédito

Capítulo 2

Marco teórico

2.1 Teoría relacionada con la problemática

El uso de la tecnología es una de las características que ha marcado el siglo XXI y ha sido de suma importancia para el crecimiento de muchos países. Está en casi todos los ámbitos de nuestra sociedad, desde un hospital hasta en cadenas de supermercados. Por ello, es prioridad crecer de acuerdo al ritmo que la sociedad impone, aquellas empresas que no se actualizan sólo les queda ser absorbidos por otras empresas o bien cerrar.

A la vez que nuestros sistemas crecen de manera exponencial yendo del bit al terabyte en un pestañeo, nuestra sociedad se ve forzada a desarrollarse a un ritmo considerable pero, a la vez que nuestro país progresa, la divergencia entre las riquezas va haciendo más evidente la diferencia entre el pobre y el rico. Es por ello que han nacido mecanismos de ayuda a la sociedad, desde programas de apoyo hasta microfinancieras.

Las microfinancieras surgen como un apoyo a las familias de escasos recursos, aquellas que no pueden acceder de manera rápida a un crédito debido a que no cuentan con un ingreso sólido o con las garantías necesarias. Estas empresas ayudan a la población de escasos recursos, sin embargo se debe contemplar que el 45.5%³ (que representa aproximadamente 53,300,000 de mexicanos) se encuentran en algún nivel de pobreza por lo que este tipo de empresas encuentran un gran campo laboral al ser un escape para muchos de los mexicanos.

La actividad más frecuente en el área de las microfinancieras es la provisión de créditos a pequeñas iniciativas autogestionadas. Se ha comprobado que tener acceso a un crédito es un factor significativamente importante que, junto con otras condiciones, puede ayudar a que millones de personas salgan de una situación de pobreza; sin embargo, aun reconociendo su importancia, existe una creciente preocupación respecto al impacto que estos programas puedan tener sobre los objetivos de desarrollo ya que representan un cambio en el propio sector financiero.

No obstante lo anterior, debe reconocerse que frente a los graves problemas de accesibilidad a los servicios financieros de todo tipo (crédito, ahorro, medios de pago), las microfinancieras aportan accesibilidad y cercanía que podrían contribuir a la construcción de sistemas financieros incluyentes y alternativos, con importantes efectos para el desarrollo.

Teniendo en cuenta el alcance de las microfinancieras en todo el mundo, es razonable pensar que estén induciendo importantes cambios en los sistemas financieros, y es necesario estudiar qué efectos pueden esperarse de ello, y qué políticas podrían

³ Según informa el Consejo Nacional de Evaluación de la Política de Desarrollo Social (Coneval) en el anexo estadístico 2012 de la pobreza en México.

<http://www.coneval.gob.mx/Medicion/MP/Paginas/Pobreza-urbana-en-M%C3%A9xico-.aspx> , Enero 2016

proponerse para que estos cambios afecten de forma positiva a objetivos de desarrollo, equidad y lucha contra la pobreza.

2.1.1 Definición de microcrédito y microfinanzas ⁴

Para saber la importancia de las microfinancieras se requiere primero conocer los conceptos básicos como microcrédito y microfinanzas.

Microcrédito es el nombre concedido a aquellos programas que otorgan préstamos pequeños a personas pobres para proyectos que son generadores de ingreso y de auto-empleo, permitiendo así el cuidado y la manutención de ellos y sus familias.

Las microfinanzas, por tanto, se refieren a los servicios financieros en reducida escala –sobre todo crédito y ahorros- proporcionados a la gente que cultiva, pesca o cría animales; que opera empresas pequeñas o microempresas donde se producen, se reciclan, reparan o venden mercancías; que proporcionan servicios; que ganan un ingreso por alquileres de pequeñas parcelas de tierra, vehículos, animales de trabajo o maquinaria y herramientas; y a otros individuos y grupos en el ámbito local de países en vías de desarrollo, tanto en áreas rurales como urbanas. Muchos de éstos tienen múltiples fuentes de ingreso.

En la “Ley de Microfinanzas” se define microcrédito como:

(...) tiene como objeto crear, estimular, promover y desarrollar el sistema microfinanciero orientado a facilitar el acceso a los servicios financieros y no financieros, en forma rápida y oportuna, a las comunidades populares y autogestionarias, las empresas familiares, las personas naturales autoempleadas o desempleadas y cualesquiera otras formas de asociación comunitaria para el trabajo, que desarrollen o tengan iniciativas para desarrollar una actividad económica, a objeto de integrarlas en las dinámicas económicas y sociales del país.

En relación a la sostenibilidad y sustentabilidad como aspecto válido quizá no sólo como finalidad a lograr en los microempresarios o público objeto de atención de la política, sino también de la gestión, aplicación y administración de los recursos que alimentan dicha política como tal.

(...) Se puede dejar de una vez establecido que hasta en la aplicación de programas crediticios por parte del Estado, en la práctica, independientemente de su naturaleza, la concepción que se maneja es más de distribución de los recursos que de inversión, por lo que es insignificante el esfuerzo que se hace por recuperar los fondos concedidos como préstamos.

⁴ Decreto con Fuerza de Ley Especial de Asociaciones Cooperativas (Gaceta Oficial N° 37.285 del 18-09-2001.)

Ante esta afirmación resulta pertinente hacer referencia al artículo 3 de la Ley de Microfinanzas, el cual estipula que, “los entes u organizaciones públicas y privadas que integren el sistema microfinanciero deben ser diligentes en el retorno y la recuperación de los recursos económicos y la sustentabilidad del sistema en forma eficiente y oportuna”.

2.1.2 Antecedentes de las microfinancieras

En 1974, el profesor Muhammad Yunus de Bangladesh propuso una forma de organización social para las aldeas rurales a las que llamó *gram sarker* (gobierno rural). De esa forma aparecen los bancos comunales y en 1976 se crea el *Grameen Bank* en Bangladesh.

Para 1984, el bank rakyat (banco estatal de desarrollo agrícola) en Indonesia introdujo los primeros servicios microfinancieros.

Antecedentes en México.

Teniendo su origen en 1990, la *Institución de Asistencia Privada* (lo que hoy se conoce como *Compartamos Banco*) inició un programa de créditos cuyo objetivo resultó similar al del banco Grameen de Bangladesh, en que se otorgaban a mujeres microempresarias bajo un esquema solidario y de aval en los estados de Oaxaca y Chiapas.

Sin embargo, a pesar de que esta institución permite el acceso a un crédito a personas que de otro modo estarían marginadas, no comparte completamente la filosofía de sus homólogas de Bangladesh en lo relativo a consolidar la propiedad del banco con los mismos acreditados o la limitación de ganancias a un tope de dividendos por año. Ello resulta lógico puesto que *Compartamos Banco* es un negocio lucrativo que busca generar rentabilidad y que fondea principalmente con recursos del mercado de valores.

Vale la pena, por otro lado, mencionar el caso del Sistema Cooperativo de Cajas de Ahorro, la cual opera en el estado de Yucatán de acuerdo a la Ley de Ahorro y Crédito Popular de 2001 y que remonta sus inicios al año de 1967. Esta institución ha ampliado su oferta de productos otorgando incluso créditos hipotecarios.

2.1.3 Importancia de las microfinancieras

Para demostrar la importancia de las microfinancieras es necesario aclarar las condiciones, requisitos y limitaciones que la población en general sufre en la actualidad para acceder a un crédito o algún tipo de ayuda, así como conocer las alternativas de financiación informal existentes y el crecimiento del sector microfinanciero.

El sector financiero informal supone un antecedente de los servicios microfinancieros actuales, y convive con otros programas más estructurados en muchos países.

A continuación se listan las limitantes más comunes:

- a) Barreras, accesibilidad y uso del sistema financiero
En cualquier sociedad hay personas que no tienen acceso a los servicios financieros formales. En los países de renta alta, se trata de un grupo minoritario con frecuencia en situación de exclusión social, pero en los países de renta media y baja es una situación habitual para gran parte de la población
 - b) Barreras económicas
La pequeña escala de las operaciones que esta población podría desarrollar no las hace interesantes para las instituciones financieras. A su vez, los precios o comisiones mínimas pueden hacer poco atractivos los productos financieros para estas personas. La falta de garantías y avales es un factor que hace imposible el crédito en muchos casos
 - c) Barreras culturales y sociales
Gran parte de la población de países empobrecidos, no es fácil ni común relacionarse con las entidades financieras formales. La dificultad es aún mayor si existen problemas de alfabetización o la necesidad de utilizar lenguas diferentes a la materna
 - d) Desde una perspectiva de género
Las dificultades son mayores para las mujeres, con menor acceso a las relaciones extra-domésticas y de negocios
 - e) Barreras prácticas
La población rural, aun siendo mayoría en muchos casos, puede tener a varias horas de distancia la oficina bancaria más cercana
 - f) Cuestiones tan triviales
Como disponer de una cédula o carnet de identidad, tener títulos de las propiedades inmobiliarias o contratos de trabajo puede ser una barrera infranqueable para muchas personas. El número de documentos necesario para abrir un depósito es también mayor en los países de menor renta.
- Préstamo mínimo de consumo como porcentaje del PIB per cápita

La información se presenta segregada por grupos de países en función de sus niveles de ingreso, de acuerdo a la clasificación del Banco Mundial

A estos problemas de acceso y de exigencias de documentación, deben añadirse las dificultades de accesibilidad derivadas de la deficiencia en la red de sucursales financieras

- Penetración geográfica
Se mide de acuerdo al número de sucursales por cada 1,000 km²
- Penetración demográfica
Se mide de acuerdo al número de sucursales por cada 100,000 habitantes

Tabla 1. Accesibilidad geográfica y demográfica por niveles de ingreso (medianas)⁵

Nivel de Ingreso	Penetración geográfica	Penetración demográfica
Ingreso alto	46.94	29.86
Ingreso medio-alto	6.27	9.69
Ingreso medio-bajo	3.27	4.7
Ingreso bajo	0.96	1.63

La Tabla 1 muestra cómo las posibilidades de acceso disminuyen significativamente en los países con ingresos menores. La diferencia contrastante que se presenta entre los países de menor renta (con menos de una sucursal por cada 1,000 km²) frente a los de ingreso más alto (más de 46 sucursales), da una idea de las diferencias de accesibilidad. En el caso de la penetración demográfica, que puede ser un indicador más apropiado al eliminar las peculiaridades geográficas, la diferencia es también importante (de 1.63 sucursales por 100,000 habitantes a más de 29, dependiendo del grupo de renta).

Como consecuencia de estos obstáculos, el acceso de los hogares al sistema financiero se ve seriamente limitado en muchas partes del mundo. Banerjee y Dufflo (2006)⁶ incluyen este aspecto en su estudio dirigido a analizar cómo se vive realmente con menos de 1 ó 2 dólares al día, recogiendo encuestas en 13 países.

En relación a los servicios financieros, muestran que el uso efectivo del sistema formal por parte de estas personas es muy reducido. El financiamiento informal es habitual, pero los costos son elevados (en el caso de Udaipur, India, donde se dispone de datos de los prestamistas, los intereses superan el 3% mensual).

⁵ Datos Banco Mundial, Beck, Kunt y Martínez Perla M.S. (<http://www.bancomundial.org/>)

⁶ <http://economics.mit.edu/files/530>, Diciembre 2015

Tabla 2. Uso efectivo y cuentas según nivel de ingreso⁷

Nivel de Ingreso	Uso efectivo (%)	No. préstamos % habitantes	No. depósitos % habitantes
Ingreso alto	87.15	449.7	2 012.82
Ingreso medio-alto	49.52	249.43	1 061.97
Ingreso medio-bajo	30.28	74.1	553.1
Ingreso bajo	19.65	21.71	120.71

Tratando de estudiar qué uso efectivo del sistema financiero se da en diferentes lugares, se han recogido en la Tabla 2 datos de estudios del Banco Mundial. Las variables estudiadas son:

- Uso efectivo. Mide el porcentaje de población adulta que tiene acceso a una cuenta en una institución financiera, calculándolo a partir de encuestas directas y los datos de intermediarios financieros. Los datos son tanto de entidades financieras y reguladoras como de instituciones microfinancieras (IMF), se dispone de información de 148 países
- Número de préstamos por cada mil habitantes. Se refiere a entidades reguladas de 44 países
- Número de depósitos por cada mil habitantes. Se refiere a entidades reguladas de 54 países. Las cuentas de depósito y préstamo por cada mil personas nos ofrecen una imagen similar, con importantes diferencias entre grupos de países. Los préstamos pasan de prácticamente 450 a 22 por cada mil habitantes en función del grupo de ingreso y los depósitos de más de 2,000 a 120

2.1.4 El financiamiento informal como alternativa

El financiamiento informal, que trata de suplir las carencias e ineficiencias del sector financiero formal, es especialmente importante en zonas rurales de Asia o África, y su importancia disminuye a medida que en cada zona se dan procesos de desarrollo y formalización de la economía.

En México, el sistema informal es utilizado por la población de bajos ingresos y, en su gran mayoría, no cuenta con supervisión y estadísticas oficiales. En este sistema, el crédito es sustituido por préstamos de familiares y amigos, tandas y agiotistas.

⁷ Datos del Banco Mundial. Honchan (2006) y Back, Kurg y Martínez Perta (<http://www.bancomundial.org/>)

De acuerdo a datos recabados por la CONDUSEF (Comisión Nacional para la Protección y Defensa de los Usuarios y Servicios Financieros) existen prestamistas prendarios que pertenecen a este sistema como los Montepíos (casas de empeño) que operan de forma totalmente legal y los cuales son la fuente de crédito más recurrida por las personas de bajos recursos.⁸

La Tabla 3 muestra un comparativo entre el financiamiento formal e informal en México.⁹

Tabla 3. Las finanzas populares en México: formal e informal

Formal	Tipo de préstamo	Informal
Crédito bancario	Crédito	Crédito de amigos y parientes Ligado a otras transacciones: <ul style="list-style-type: none"> • Con el empleo • Con la renta de la tierra • Con compras de insumos/ventas de productos Microempresa rural <ul style="list-style-type: none"> • “Coyotes” • “Acaparadores” • Empresarios rurales de tiendas de abarrotes Otras <ul style="list-style-type: none"> • Compra a plazos a tiendas urbanas • Fonacot • De tiendas semiurbanas • De aboneros • Agiotistas profesionales • Nacional Monte de Piedad, Montepío Luz Saviñón Prestamistas prendarios informales
Depósitos bancarios Cuentas de ahorro para el retiro(Sistema de Ahorro para el Retiro, SAR) Banco del Ahorro Nacional y Servicios Financieros (BANSEFI)	Servicios de ahorro	<ul style="list-style-type: none"> • Animales, joyería, etc. • Atesoramiento de efectivo • Préstamos a terceros
Sociedades de ahorro y préstamo Uniones de crédito	Mecanismos de ahorro y crédito	Caja Popular Mexicana, A.C. Cajas independientes Cajas operadas por compañías para sus empleados y por sindicatos para sus afiliados Cajas solidarias
Giros postales, cheques de caja y money orders Transferencias electrónicas Cambio de cheques. Pago en bancos de teléfono, electricidad, agua y predial	Servicios de pagos	Transferencia de bolsillo Cambio de cheques negocios no bancarios

⁸ <http://www.condusef.gob.mx/index.php/otros-servicios-y-productos/945.html> Marzo 2015

⁹ Mansell Carstens, Catherine. *Las finanzas populares en México. El redescubrimiento de un sistema financiero olvidado*. CEMLA. Milenio, ITAM. México, 1995. Pagina 74.

2.1.5 Sistema financiero mexicano

El sistema financiero es aquel que se encarga del funcionamiento y desarrollo de la economía. Está integrado principalmente por diferentes intermediarios y mercados financieros, a través de los cuales una variedad de instrumentos movilizan el ahorro hacia sus usos más productivos. Los bancos son conocidos como aquellos intermediarios financieros más usados y conocidos, puesto que ofrecen directamente sus servicios al público y forman parte medular del sistema de pagos. Sin embargo, en el sistema financiero participan muchos otros intermediarios y organizaciones que ofrecen servicios de gran utilidad para la sociedad.

Los sistemas financieros se caracterizan o se espera sean estables, eficientes, competitivos e innovadores lo cual contribuye a elevar el crecimiento económico de una población. Para ello se cuenta con un marco institucional sólido y una regulación y supervisión financieras que guarda la integridad del mismo sistema y protejan los intereses del público.

El esfuerzo del sistema por hacer accesibles los servicios financieros superando los métodos informales más tradicionales (familia, prestamistas, etc.) no es el primero históricamente, no obstante ha resultado el más exitoso en términos de alcance y número de clientes, desplazando otras posibles vías de mayor implicación pública en éste.

2.1.6 Características del sistema financiero mexicano¹⁰

La principal función de un sistema financiero es ser el intermediario entre quienes tienen y quienes necesitan dinero. La ventaja para aquellos que tienen dinero y desean obtener un premio a cambio de sacrificar el beneficio inmediato es la *tasa de interés*.

La tasa de interés es el costo adicional que están dispuestas a pagar las personas que requieren, en un plazo corto de tiempo, más dinero del que poseen, ya sea para generar un valor agregado mediante la realización de un proyecto o bien para cubrir una obligación de pago. Para realizar el pago de este costo adicional, se fija previamente un periodo y un plan de pagos.

¹⁰ De acuerdo a lo publicado por BANXICO en:
<http://www.banxico.org.mx/divulgacion/sistema-financiero/sistema-financiero.html>, Marzo 2015

2.1.7 Descripción del sistema financiero mexicano

El sistema financiero mexicano puede dividirse como se describe en la Tabla 4:

Tabla 4. Divisiones del sistema financiero

Tipo de banca	Descripción
Banca comercial	Normalmente se encarga de atender las necesidades financieras de grandes empresas y de sectores de población con ingresos más grandes
Banca social	En su mayoría no regulada Se encarga de atender a microempresas y sectores de menores ingresos
Banca de desarrollo	Tiene la intención de fomentar el desarrollo de sectores de la economía determinados a través de créditos No tiene fines de lucro

El sector está conformado por:

- **Uniones de Crédito**¹¹

Son instituciones financieras reguladas y están organizadas como sociedades anónimas. Se encuentran bajo la supervisión y vigilancia de la Comisión Nacional Bancaria y de Valores (CNBV) y fueron creadas con el propósito de ofrecer financiamiento a sus socios

Las Uniones de Crédito tienen autorización para operar exclusivamente con sus socios, quienes tienen que cumplir los requisitos establecidos en la Ley de Uniones de Crédito para serlo, además de que deben adquirir determinado número de acciones de la Unión de Crédito que representen un equivalente a 2,500 unidades de inversión

- **Sociedades Financieras de Objeto Limitado (SOFOL)**¹²

Son instituciones financieras autorizadas por la Secretaría de Hacienda y Crédito Público (SHCP), reguladas por el Banco de México y supervisadas por la CNBV. Su objetivo es otorgar créditos o financiamientos para la planeación adquisición, desarrollo y administración de todo tipo de bienes (muebles e inmuebles). Está

¹¹ De acuerdo a la CONDUSEF

<http://www.condusef.gob.mx/index.php/instituciones-financieras/otros-sectores/uniones-de-credito>, Marzo 2015

¹² De acuerdo a la CONDUSEF

<http://www.condusef.gob.mx/index.php/instituciones-financieras/otros-sectores/sociedades-financieras-de-objeto-limitado-sofoles>, Marzo 2015

dirigido a actividades y sectores específicos, aquellos que no han tenido acceso a los créditos ofrecidos por los sistemas financieros tradicionales

- **Sociedades Financieras de Objeto Múltiple (SOFOM)¹³**

Son entidades financieras que otorgan créditos a diversos sectores y que obtienen sus recursos mediante el fondeo en instituciones financieras y/o emisiones públicas de deuda. Una SOFOM no puede captar recursos del público y no necesita autorización del gobierno federal para constituirse

Estas sociedades gozan de las mismas ventajas fiscales y procesales que tenían al ser SOFOL, arrendadoras y empresas de factoraje pero con las ventajas civiles de ser empresas mercantiles

Existen dos clases de SOFOM: reguladas y no reguladas:¹⁴

Las SOFOM reguladas están bajo la supervisión de la CNBV y, de acuerdo a la *Ley general de organizaciones y actividades auxiliares de crédito*, son aquellas que *mantengan vínculos patrimoniales con instituciones de crédito o sociedades controladoras de grupos financieros de los que formen parte las instituciones de crédito*. Estas sociedades deben incluir en su denominación social la leyenda “sociedad financiera de objeto múltiple” o “SOFOM”, en su defecto, seguido de “entidad regulada” (o “E. R.”)

Las SOFOM no reguladas son igualmente supervisadas por la CNBV aunque únicamente en temas respectivos a lavado de dinero. Estas sociedades son aquellas en que no tengan participación cualquiera de las entidades mencionadas en el punto anterior en su capital. Al igual que las SOFOM reguladas, las no reguladas deben agregar a su denominación social la leyenda “sociedad financiera de objeto múltiple” o “SOFOM”, seguido de “entidad no regulada” (o “E. N. R.”)

- **Sociedades Cooperativas de Ahorro y Préstamo (SCAP)¹⁵**

La *Ley para Regular las Actividades de las Sociedades Cooperativas de Ahorro y Préstamo* define una SCAP como una sociedad cuyo objetivo es realizar operaciones de ahorro y préstamo con sus socios reconociendo que no son intermediarios financieros con fines de lucro.

¹³ De acuerdo a la CONDUSEF

<http://www.condusef.gob.mx/index.php/instituciones-financieras/sociedades-financieras-de-objeto-multiple>, Marzo 15

¹⁴ De acuerdo a la CONDUSEF

<http://www.condusef.gob.mx/index.php/instituciones-financieras/64-sociedades-financieras-de-objeto-multiple/633-caracteristicas-de-una-sofom>, Marzo 2015

¹⁵ De acuerdo a la CONDUSEF

<http://www.condusef.gob.mx/index.php/instituciones-financieras/sociedades-cooperativas-de-ahorro-y-credito-popular/916-sociedades-cooperativas-de-ahorro-y-prestamo-scap>, Marzo 2015

La autorización y el tipo de operaciones de las SCAP dependen del monto total de los activos con que cuenten las mismas (establecidas en unidades de inversión, UDIS) y, de acuerdo al monto del préstamo, se establece un nivel de operación.

Nivel I: monto de activos totales \leq 10 millones de UDIS

Nivel II: 10 millones < monto de activos totales \leq 50 millones de UDIS

Nivel III: 50 millones < monto de activos totales \leq 250 millones de UDIS

Nivel IV: monto de activos totales > 250 millones de UDIS

Cada uno de estos niveles requiere de la autorización de la CNBV y cuentan con un fondo de protección por hasta 25 mil UDIS por socio

- **Sociedades Financieras Populares (SOFIPO)**¹⁶

Son entidades de microfinanzas constituidas como sociedades anónimas de capital variable que operan bajo la supervisión de la CNBV. Dichas instituciones tienen facultad para prestar servicios tanto a socios como a clientes, de acuerdo a lo establecido en la *Ley de Ahorro y Crédito Popular*

- **Sociedades Financieras Comunitarias (SOFINCO)**¹⁷

Es una institución financiera integrada por personas físicas y morales dedicada a prestar servicios financieros a sus socios y terceros en zonas rurales. Está regulada por la *Ley de Ahorro y Crédito Popular*

Legalmente se constituye como una sociedad anónima bajo límites de propiedad colectiva y base democrática (ninguna persona física puede poseer más del 1% del capital social)

¹⁶ De acuerdo a la CNBV

<http://www.cnbv.gob.mx/SECTORES-SUPERVISADOS/SECTOR-POPULAR/Preguntas-Frecuentes/Paginas/Sociedades-Financieras-Populares.aspx>, Marzo 2015

¹⁷ <http://sofinco.com.mx/images/Figuras.pdf>, Marzo 2015

2.1.8 Plan de administración de una microfinanciera

Para el correcto funcionamiento de una microfinanciera es necesario realizar una planificación táctica para gestionar la infraestructura y los activos de la organización. Sin embargo el plan de administración no finaliza con la definición de fases, actividades, responsables, tiempos, etcétera, sino que es necesaria una gestión que asegure el logro de cada uno de los objetivos planteados.

Los aspectos básicos a considerar para una adecuada administración son:

- Estructura organizacional, encargada de dirigir el plan de administración
- Estrategias de comunicación y administración de riesgos
- Metodología de seguimiento

Estructura organizacional

Independientemente de las dimensiones que posea la organización, se podrían distinguir ciertos papeles principales dentro de ella, como se muestra en la Tabla 5.

Tabla 5. Estructura organizacional de una microfinanciera

Rol	Descripción
Director comercial	Es la persona que se encarga de realizar la promoción y difusión de los servicios, así como de dar atención a los clientes
Director operativo	Es la persona que tiene a cargo la administración de la microfinanciera. Del mismo modo, se encarga que los procesos internos de la empresa se lleven a cabo apropiadamente
Asesores	Ellos se encargan de ofrecer un servicio personalizado a los clientes, explicándoles detalladamente los procesos que deben llevarse a cabo desde la adquisición del crédito hasta la forma en que deben realizarse los pagos del mismo Igualmente, se encargan de realizar una evaluación con la que sea posible determinar el monto que puede prestarse a cada solicitante

Organigrama de una microfinanciera

La composición del organigrama considera tanto el alcance de los objetivos así como a los responsables del cumplimiento de cada uno de ellos.

En la Figura 4 se muestra el organigrama básico de una microfinanciera:

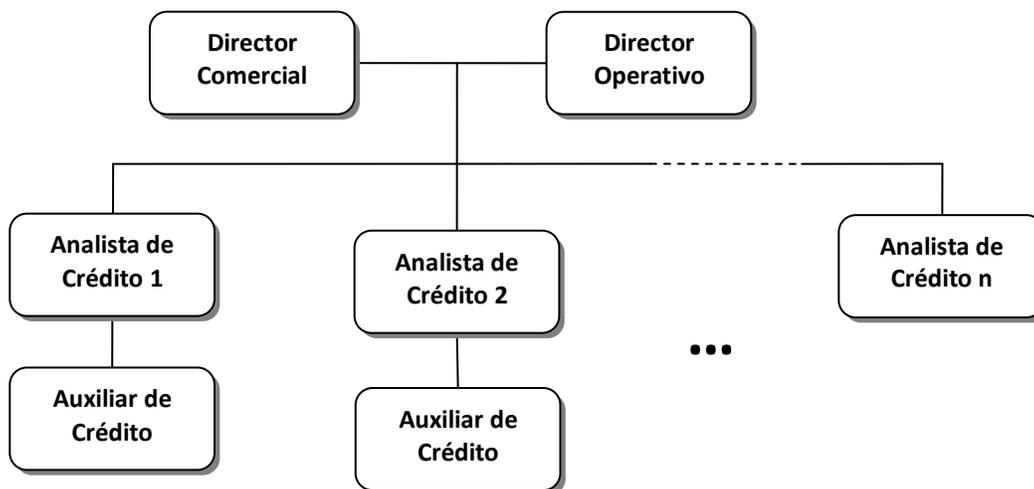


Figura 4. Organigrama de una microfinanciera

Es importante realizar reuniones cada cierto periodo de tiempo establecido, con la finalidad de garantizar el correcto funcionamiento de las tareas que le han sido asignadas a cada rol. Esto ayudará, del mismo modo, a detectar fallas en el sistema utilizado y corregirlas lo antes posible, asegurando así la calidad de los servicios prestados por la microfinanciera.

Estrategias de comunicación y administración de riesgos

Las estrategias de comunicación hacen referencia a la forma en la cual los directores dan a conocer sus metas a sus empleados, así como cada punto de crecimiento se exhibe para mejorar.

Un riesgo es toda eventualidad que puede provocar la interrupción parcial o total de un proyecto, por ello es indispensable contar con la identificación de cada uno de los riesgos. Con ello se puede contar con un mayor control de cada uno de estos eventos, logrando administrarlos y evitarlos.

2.1.9 Tecnologías microcrediticias

Las instituciones de microfinanzas desarrollaron tecnologías para la atención especializada de prestatarios a pequeña escala, los cuales suelen carecer de registros contables y/o garantías y, en la mayoría de los casos, incluso de formalización. Es precisamente por ello que les resulta difícil a los bancos cubrir las necesidades de estos estratos de la población.

A continuación se muestra en qué consiste la tecnología del microcrédito:

- Elaborar los estados financieros de los clientes con la finalidad de evaluar la unidad de riesgo de acuerdo a los ingresos-egresos de cada uno, y con ello calcular la capacidad de pago, fijar el monto del crédito y adecuar el calendario de pagos de acuerdo a los ingresos del cliente
- Estimar la voluntad de pago del cliente a través de los datos recabados en su entorno socioeconómico. Esta información se consigue de vecinos, proveedores, clientes, etcétera
- El monto del crédito es calculado a partir de una proporción de los activos y pasivos (patrimonio) y el calendario de pagos
- El asesor del crédito es el encargado de presentar la propuesta al comité de la sucursal
- Los clientes que paguen puntualmente pueden tener un trato preferencial que les permita nuevos préstamos en el futuro por montos cada vez mayores, aunque cuidando no rebasar nunca sus posibilidades de pago

Grupos solidarios y crédito individual

Los grupos solidarios son una modalidad donde las personas que tienen negocio se juntan en grupos pequeños con la finalidad de obtener un crédito y garantizarse mutuamente.

Los créditos individuales son aquellos que se otorgan a un cliente cumpliendo con los requerimientos necesarios. En la Tabla 6 se podrá ver la comparación de los grupos solidarios y créditos individuales.

Tabla 6. Comparativa de grupos solidarios y crédito individual¹⁸

Tecnología grupal	Tecnología individual
A nivel de los clientes	
<ul style="list-style-type: none">• Crédito condicionado a que los clientes se garanticen mutuamente• Los miembros del grupo deben respaldarse y apoyarse• La garantía solidaria evita las garantías	<ul style="list-style-type: none">• Acepta diversas garantías, incluso prendas simbólicas• Al carecer de grupo, los clientes se evitan costos de organización y mantenimiento, por

¹⁸ <http://www.financiarural.gob.mx/ApoyosProductoresIntermediarios/Documents/Julio%202011/Modelo%20T%C3%A9cnico%20Financiero%20SOFOM/5%20Manual%20de%20Credito.pdf>, Abril 2015

reales, lo cual es útil para personas que iniciarán un negocio	lo que responden únicamente por sus deudas <ul style="list-style-type: none"> • Los montos y los plazos de pago se ajustan a sus necesidades y capacidades
A nivel de las instituciones	
<ul style="list-style-type: none"> • Evaluar a los clientes es más sencillo ya que es el mismo grupo quien lo hace • El costo administrativo por prestatario es menor • Los costos de transacción son menores 	<ul style="list-style-type: none"> • Se conoce mejor a los clientes debido a que, al ser individuales las evaluaciones, se detalla su actividad y capacidad de pago • Se detecta a los mejores clientes sin necesidad de buscarlos en el grupo

2.1.10 Ciclo de un crédito

Para la apertura de un crédito necesita llevarse a cabo una serie de pasos de manera ordenada y cuidadosa. A continuación se describe cada uno de los pasos considerando los siguientes parámetros:

- Responsable principal. Aquel que es el encargado de realizar la operación
- Acciones de promoción. Aquellas acciones que deben realizarse durante esa etapa
- Referidos. En qué consisten las acciones

Ciclo de un crédito

a) Promoción del crédito (Figura 5)

- 1) Analista de créditos o promotor
- 2) Conocer la estrategia para la promoción y la captura de acreditados. Las vías usuales de ingreso (publicidad en medios masivos: prensa escrita, radio, televisión, etc.)
- 3) Metodología aplicada de promoción, que consiste en contactar con personas que ya han operado crediticiamente con buen historial y que en la actualidad no cuentan con crédito activo o, solicitar información sobre acreditados potenciales hablando con personas como fuentes de información (jefes de entidades gremiales, administradores de mercados, proveedores de negocios minoristas)

Promoción directa. La promoción directa o activa es aquella que realiza directamente el analista de créditos y también, los gerentes de sucursales o personal auxiliar de créditos visitando a los potenciales acreditados en su lugar de desarrollo de la actividad generadora de ingresos.

Solicitante potencial. Todas las acciones de promoción tienen como finalidad, la detección de solicitantes potenciales.

Se informa sobre producto. El personal de la Institución que ha logrado captar la atención del solicitante potencial, informará al mismo sobre la gama de productos ofertados por la Institución y sus características.

Interés. Expuestas las ventajas y beneficios de operar, el solicitante estará en condiciones de tomar la decisión de optar o no por un financiamiento.

Llenado de formulario. Pre-solicitud: si el solicitante potencial cuenta con un buen historial de pago, o se trata de un potencial nuevo acreditado, se procederá al llenado de la pre-solicitud en donde se toman los datos básicos de la persona (nombre completo, dirección, teléfono, referencias personales y comerciales, monto solicitado, plazo, destino de la inversión, etc.).

La toma de datos de la pre-solicitud es el primer paso en el proceso de concesión de créditos y no necesariamente implica realizar una visita al acreditado.

El objetivo principal en esta etapa es conseguir información básica sobre el acreditado y su actividad generadora de ingresos, la cual servirá como punto de partida para iniciar la evaluación del riesgo.

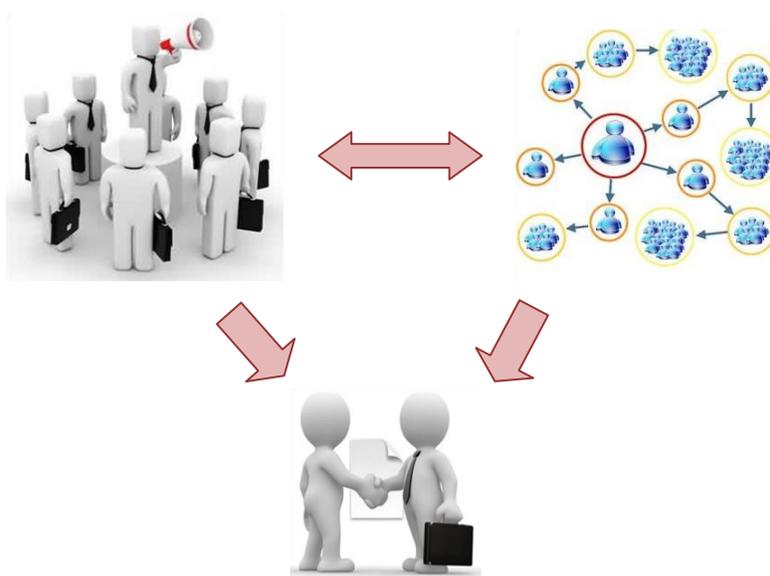


Figura 5. Promoción del crédito

b) Consulta en sociedades de información crediticia (Figura 6)

- 1) Auxiliar de créditos
- 2) Se debe verificar las referencias tanto del solicitante, su cónyuge/conviviente, del aval y su cónyuge/conviviente (de ser el caso)

- 3) Auxiliar de crédito recibe formulario: el auxiliar de crédito procederá a la recepción de las pre-solicitudes que hayan sido llenadas por analistas de crédito en campo, a fin de realizar la correspondiente consulta

Consulta base de datos. El auxiliar de crédito o, en su defecto el gerente de sucursal, procederá a la consulta e impresión del reporte del solicitante.

Antecedentes negativos. El auxiliar de crédito interpretará el informe emitido por las sociedades de información crediticias y de la base de datos institucional.

Asigna analista de crédito. El expediente del acreditado potencial, es entregado al analista de crédito, quien se encargará de realizar la visita de evaluación al domicilio.



Figura 6. Consulta en sociedades de información crediticia

c) Control de las pre-solicitudes

- 1) Auxiliar de créditos
- 2) Control de pre-solicitudes. El registro tiene como objetivo permitir el seguimiento y control de las solicitudes, y tener información sobre quién es el responsable de la misma para responder las preguntas de los solicitantes y para acelerar el proceso si fuera necesario
- 3) Existen pre-solicitudes pendientes. Se generará en el sistema informático de presentarse el caso, y existan pre-solicitudes pendientes de asignación a analista de crédito

Verificar retraso con analista de crédito asignado. Pre-solicitudes que aún no hayan sido atendidas por el analista de crédito correspondiente, el auxiliar de crédito procederá a informarse de los motivos por los cuales, dicha solicitud de financiamiento aún no ha sido atendida.

Visita de llenado de solicitud y análisis. Llenado de los formularios (solicitud de crédito, evaluación de riesgos y declaración patrimonial).

d) *Visita de Evaluación al Solicitante*

- 1) Analista de crédito
- 2) Visita de llenado de formulario y análisis. Se completará la información correspondiente en los formularios para la formalización de la solicitud de crédito, según lo siguiente: Si es acreditado nuevo o no, tipo de persona, producto crediticio solicitado
- 3) Al firmar la solicitud de crédito y entregar toda la documentación correspondiente, el solicitante declara la veracidad de los datos presentados y formaliza su pedido. Los datos que el analista de crédito ha recopilado, serán registrados directamente en el sistema informático para su procesamiento y control

Recopilación y elaboración de estados financieros tiene dos objetivos:

- Determinar la capacidad de pago

La capacidad de pago es el aspecto cuantitativo del análisis, por lo tanto se puede determinar con cierta precisión, si el analista de crédito tiene la suficiente habilidad para tomar los datos correctamente y si el acreditado es transparente al dar la información

- Determinar la moral de pago

La moral de pago es el aspecto cualitativo/subjetivo, porque depende de la apreciación que tenga el analista de crédito sobre el comportamiento del acreditado, su entorno y su familia. El objetivo de esta evaluación es determinar si el acreditado tiene pre-disposición para cumplir puntualmente con sus obligaciones

Durante el primer contacto se considera apto. El analista de crédito deberá analizar la viabilidad de la operación crediticia. Se le informa al solicitante, da de baja y archiva en rechazados.

El objetivo en esta etapa es recoger la opinión que tienen las personas sobre el solicitante para establecer su moral de pago, tratando siempre de obtener información adicional que podría ser útil en el futuro.

Procesamiento de información financiera. Analiza los datos recolectados y completa el expediente con los documentos.

Cumple requisitos para presentación en comité de crédito: Con la información recopilada, procesada y analizada, el analista de crédito está en condiciones de establecer si el solicitante potencial cumple con los requisitos.

Expediente para presentación en comité de crédito: Si con base en la información procesada y analizada, el analista de crédito determina que el solicitante cuenta con capacidad de pago y las garantías requeridas según lo establecido en el manual de crédito, deberá elaborar su propuesta de financiamiento para su presentación en comité de crédito.

e) Comité de Crédito

- 1) Miembros del comité según el nivel de aprobación
- 2) Expediente para presentación comité de crédito. Se procederá a realizar la verificación de la información proporcionada por los analistas de crédito en función a las pautas establecidas, que puede ser según montos de crédito

Auxiliar de crédito controla expediente. El auxiliar de crédito controlará todos los expedientes que se encuentran listos para ser presentados en comité de crédito con la finalidad de verificar si el expediente de la operación en proceso cuenta con toda la documentación y formularios requeridos

- 3) Expediente completo. Cada expediente deberá contar con la documentación respaldatoria de la operación crediticia, la cual dependerá del tipo de producto, monto solicitado y garantías en caso de no cumplir se deberá devolver al analista de crédito y deberá regularizar la documentación

Se comunica al solicitante. La comunicación con el acreditado potencial en forma inmediata al término de la sesión del comité de crédito, a fin de informar al mismo los cambios que se han realizado en el comité de crédito y realizar las correspondientes negociaciones.

El auxiliar de crédito formaliza la operación. El auxiliar de crédito registrará las operaciones aprobadas en el sistema informático, tanto las aprobadas sin modificaciones como aquellas que fueron modificadas y aceptadas por los acreditados.

f) Formalización y desembolso del crédito

- 1) Auxiliar de crédito, contabilidad, cajero
- 2) Formalización de la operación. Se continúa con el procedimiento de formalización y posterior desembolso

Auxiliar de crédito verifica el cumplimiento de las disposiciones y verificará que el expediente esté completo y procederá a ingresar al sistema informático las condiciones de la operación aprobadas por el comité de crédito

3) Preparación e impresión de documentación. La documentación a ser impresa estándar será: Contrato, pagaré, plan de pagos y cheque. Se informa al solicitante operación para desembolso

Formalización y firma de documentación. El desembolso del crédito se efectúa al concurrir el acreditado a la sucursal.

Posteriormente, completará el expediente de la solicitud haciendo firmar el pagaré respectivo al acreditado y su cónyuge/conviviente y aval y su cónyuge/conviviente si corresponde, verificando las firmas del contrato de crédito, pagaré, hoja de recepción conforme de efectivo o cheque.

Se archiva la documentación y la custodia del expediente del acreditado conteniendo: solicitud de crédito, informes del crédito, documentos que sustentan el análisis, propuesta de crédito del analista de crédito (Figura 7).

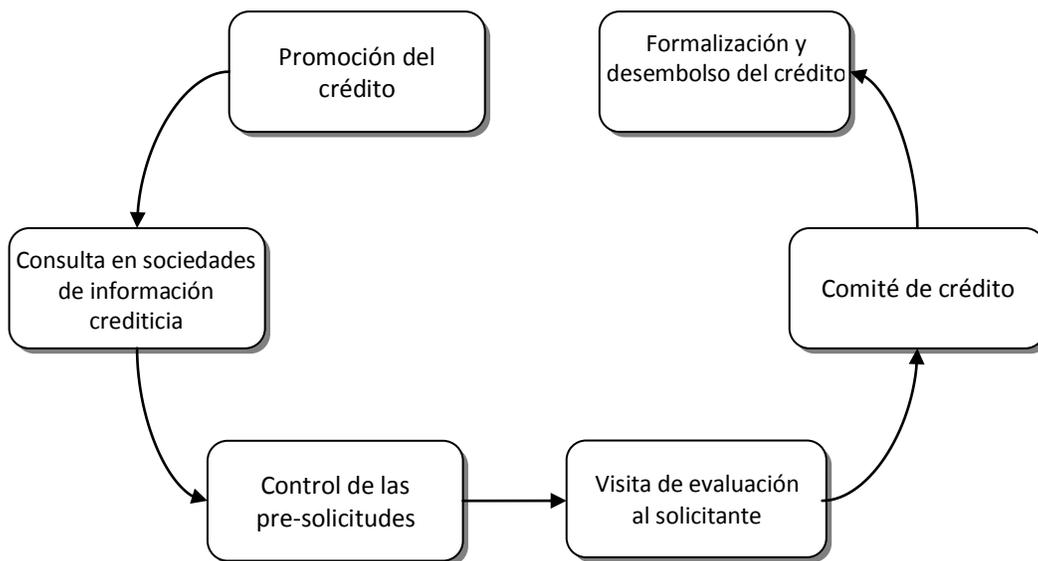


Figura 7. Ciclo del crédito

2.2 Teoría relacionada con la ingeniería de software

2.2.1 Marco teórico. La ingeniería de software

La ingeniería de software surge en 1968 durante una conferencia de la OTAN. Nace como consecuencia de la crisis de software y con la finalidad de poder desarrollar software de alta calidad, barato, con capacidad de mantenimiento y rápido de construir.

Según The Institute of Electrical and Electronics Engineers, IEEE (Instituto de Ingenieros Eléctricos y Electrónicos), es *la aplicación de un enfoque sistemático, disciplinado y*

cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software; es decir, la aplicación de ingeniería al software.

Los principales componentes de la ingeniería de software son:

- Procesos: Son un marco de trabajo que ayuda a controlar la gestión del proyecto y las actividades de ingeniería
- Métodos: Son las actividades técnicas requeridas para la creación de productos de trabajo
- Herramientas: Que ayudan a la automatización para los procesos y los métodos

El principal componente de la ingeniería de software es el proceso, ya que es la etapa donde se identifican las tareas de la ingeniería que se deben realizar para construir software de alta calidad, sin embargo la capa que nos interesa para la planeación de nuestro proyecto es la metodología.

La metodología se encarga de las actividades técnicas que se requieren para conseguir las tareas, es decir, indica el cómo podemos realizarlas.

Las actividades fundamentales para el desarrollo de software (Figura 8) son:

- Análisis: Se crea el modelo de lo que es requerido por el software
- Diseño: Traduce el análisis en cómo se puede realizar el proceso según el software
- Codificación: Traduce el modelo del diseño en una forma ejecutable
- Pruebas: El proceso de pruebas ayuda a encontrar errores en el código y el diseño subyacente

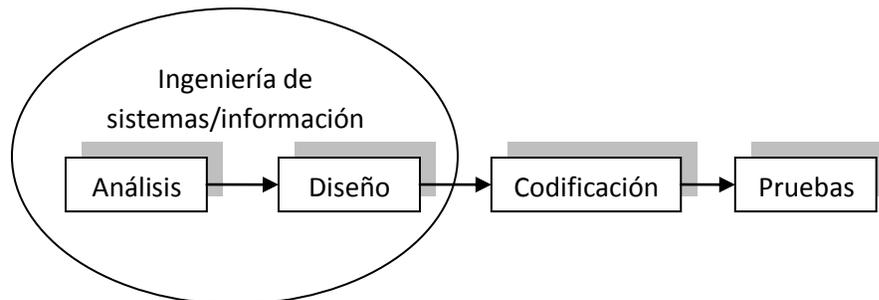


Figura 8. Actividades fundamentales para el desarrollo de software¹⁹

Podrían añadirse dos fases más a este proceso²⁰:

¹⁹ Pressman S. Roger. *Ingeniería de software. Un enfoque práctico*. McGrawHill, 5th edición, México, 2002. Pág. 20

- **Implantación:** El proceso de instalar la aplicación e instruir a los usuarios en el uso de la misma, así como “construir” la infraestructura (datos, archivos, etc.) necesaria para su funcionamiento
- **Mantenimiento:** El proceso de mejora y optimización del software después de que éste ha sido entregado al usuario final. Considera también correcciones y prevención de defectos

Por otro lado, dentro de la evolución de los procesos, se identifican cuatro actividades principales:

1. **Especificación del software:** Involucra funcionalidades, requisitos y restricciones del sistema
2. **Desarrollo del software:** Abarca tanto la planeación como el desarrollo del sistema, la creación de interfaces, codificación, etcétera, cuidando el cumplimiento de los requisitos previamente establecidos
3. **Validación del software:** Se realiza la verificación del cumplimiento de los objetivos
4. **Evolución del software:** Se refiere al mantenimiento que se le dará al sistema, es decir, la adaptabilidad del mismo con el paso del tiempo

2.2.2 Metodologías para desarrollo de software

La metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades de un proyecto en desarrollo, es un proceso de software detallado y completo.

2.2.2.1 Modelo en cascada modificado

Es uno de los métodos más utilizados en ingeniería de software. También se le conoce como *modelo secuencial o lineal* o “*modelo clásico*” haciendo una comparación con el agua de una cascada que no cambia su flujo natural que va de arriba hacia abajo.

Este modelo implica necesariamente un enfoque sistemático de modo que cada una de sus partes se relaciona entre sí para alcanzar un fin común. Del mismo modo, debe ser secuencial, lo cual implica que cada etapa depende totalmente de la información de la anterior; sin embargo, el modelo en cascada modificado permite regresar a una etapa anterior en caso de ser necesario (como puede apreciarse en la figura 9).

²⁰ <http://dicyg.fi-c.unam.mx:8080/lalo/aed-teo/FasesDS.pdf>, Diciembre 2014

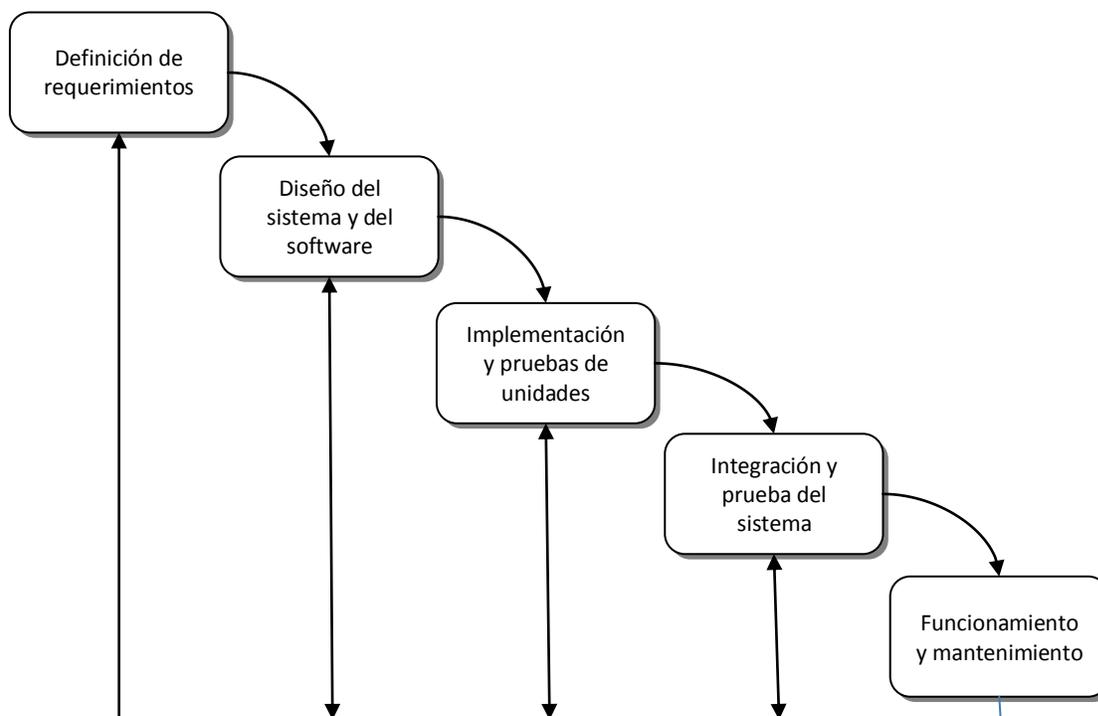


Figura 9. Modelo en cascada modificado²¹

Las etapas principales de este modelo se transforman en actividades fundamentales del desarrollo:

Definición de requerimientos

El funcionamiento del sistema como son el alcance, las restricciones y las metas dependen de lo que requieran los usuarios, por lo cual deben ser definidos a detalle.

Sin embargo, debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software para lo cual se requiere de un análisis del sistema.

El análisis se encarga de determinar los objetivos y límites del sistema, caracterizar su estructura y funcionamiento y definir las directrices que permitan alcanzar los objetivos.

Dependiendo de los objetivos del análisis, podemos encontrarnos ante dos problemáticas distintas:

²¹ Sommerville, Ian. Ingeniería del software. Ed. Pearson, 7ª edición, Madrid, 2005. Páginas 62-63.

- Análisis de un sistema ya existente para comprender, mejorar, ajustar y/o predecir su comportamiento
- Análisis como paso previo al diseño de un nuevo sistema-producto

En cualquier caso, podemos agrupar más formalmente las tareas que constituyen el análisis en una serie de etapas que se suceden de forma iterativa hasta validar el proceso completo:

Conceptualización

Consiste en obtener una visión de muy alto nivel del sistema, identificando sus elementos básicos y las relaciones de éstos entre sí y con el entorno.

Análisis funcional

Describe las acciones o transformaciones que tienen lugar en el sistema. Dichas acciones o transformaciones se especifican en forma de procesos que reciben unas entradas y producen unas salidas.

Análisis de condiciones (o constricciones)

Debe reflejar todas aquellas limitaciones impuestas al sistema que restringen el margen de las soluciones posibles. Éstas se derivan a veces de los propios objetivos del sistema:

- Operativas. Como son las restricciones físicas, ambientales, de mantenimiento, de personal, de seguridad, etcétera
- De calidad. Como fiabilidad, mantenibilidad, seguridad, convivencia, generalidad, etcétera

Sin embargo, en otras ocasiones las constricciones vienen impuestas por limitaciones en los diferentes recursos utilizables:

- Económicos. Reflejados en un presupuesto
- Temporales. Que suponen unos plazos a cumplir
- Humanos. Referentes a la cantidad de personal
- Metodológicos. Que conllevan la utilización de técnicas determinadas
- Materiales. Como espacio, herramientas disponibles, etc.²²

Análisis de los requisitos del software

El proceso de recopilación de los requisitos se centra especialmente en el software. El ingeniero de software (analista) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.

²² Sommerville, Ian. Ingeniería del software. Ed. Pearson, 7ª edición, Madrid, 2005.

Análisis de riesgos

Al contar con un sistema informático nos encontramos susceptibles a peligros tanto informáticos como físicos, es por ello que debemos tener identificados los puntos que pueden ser vulnerables.

Los puntos a los cuales se debe prestar mayor atención son aquellos que pongan en peligro la integridad, confidencialidad, disponibilidad e irrefutabilidad del sistema.

Los riesgos de integridad, son aquellos riesgos asociados con la autorización, completitud y exactitud de la entrada, procesamiento y reportes de las aplicaciones utilizadas en una organización.

- Interfaz de usuario:
 - Cada perfil cuenta con características específicas
 - El ejecutivo cuenta con las posibilidades de ofrecer y capturar y/o modificar las cuentas, clientes y créditos
 - El analista de crédito puede autorizar un crédito en caso de ser una situación excepcional
- Procesamiento:
 - El proceso de captura debe quedar respaldado en un repositorio, al cual podrá accederse en cualquier momento así como también podrá modificarse sin poseer redundancia de la información
- Interface:
 - Garantizar que la información se encuentra respalda de manera correcta en la base de datos. Se debe validar la congruencia y los datos mismos capturados que van a guardarse en la base

Riesgos de relación, son aquellos que se relacionan directamente con la toma de la información.

- Al cumplir con los requisitos necesarios para la aceptación inmediata de un crédito, éste deberá actualizarse una vez que esté autorizado sin requerir un analista de créditos
- Al capturar correctamente los datos de alta de cuentas, cliente y crédito se deberá visualizar. En caso contrario, se mostrarán los errores ocurridos en dicho proceso

Riesgos de acceso, se enfocan en el inapropiado acceso al sistema, datos e información.

- Se otorgarán los perfiles y permisos correctos a las aplicaciones

- Riesgos de utilidad: Riesgos que pueden ser enfrentados por el direccionamiento de los sistemas antes de que los problemas ocurran, minimizar la ruptura de los sistemas y backups y planes de contingencia
- Garantizar la posible modificación del sistema sin una afectación del negocio

Riesgos en la infraestructura, se refieren a que en las organizaciones no existe una estructura de información tecnológica efectiva.

- Al no contar con un área específica de redes, se hace la recomendación como un punto de oportunidad de crecimiento
- Otro punto que debe considerarse son aquellos riesgos como los delitos o fraudes informáticos

Se deberá contar con un sistema físico capaz de sostener los ataques como los virus informáticos, asegurar los puertos de entrada de dispositivos externos así como disponer de un software con firewalls lo suficientemente fuertes como para soportar todo tipo de ataque conocido.

Tener un sistema e instalaciones eléctricas, con equipo de características mínimas para soportar el número de equipos y personal requeridos en la financiera.

Diseño del sistema y del software

El objetivo del diseño es producir un modelo o representación que tenga resistencia, funcionalidad y belleza.

La fase de diseño divide los requerimientos en sistemas hardware o software y se caracteriza por considerar un conjunto de principios, conceptos y prácticas que llevan al desarrollo de un sistema o producto de alta calidad.

El diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

Lineamientos y atributos de la calidad de software

Se sugiere las tres características que funcionan como guía para evaluar un buen diseño:

- Se debe implementar todos los requerimientos explícitos
- Debe ser una guía legible y comprensible para quienes desarrollan, prueban y corrigen
- Debe proporcionar el panorama completo de software

Para lograr cumplir con estos puntos es necesario seguir las siguientes recomendaciones:

- Lineamientos de calidad. Debe tener una arquitectura que se haya creado con el empleo de estilos o patrones arquitectónicos. Del mismo modo, debe estar compuesta de componentes con buenas características de diseño y que se implementen de forma evolutiva
- Debe ser modular. Es decir, el software debe estar dividido de manera lógica en elementos
- Debe contener distintas representaciones de datos, arquitectura, interfaces y componentes
- Debe conducir a estructuras de datos coherentes para las clases que se van a implementar
- Debe llevar componentes con características funcionales independientes

Atributos de calidad

Hewlett Packard desarrolló un conjunto de atributos de la calidad de software a los que dio el acrónimo FURPS (Funcionalidad, Usabilidad, Confiabilidad, Rendimiento y Mantenibilidad)

Funcionalidad. Son todas las características y capacidades del programa, la generalidad de las funciones que se esperan y la seguridad.

Usabilidad. Se considera la constancia, la estética y la documentación.

Confiabilidad. Se evalúa con la frecuencia de fallas y la gravedad de éstas, así como su capacidad de recuperación.

Rendimiento. Es la velocidad de procesamiento, eficiencia y el tiempo de respuesta.

Mantenibilidad. Es la capacidad del software para ser ampliada, adaptable y servicial.

Diseño de arquitectura

Si una arquitectura de software resulta deficiente en su diseño, difícilmente se cumplirá con los requerimientos establecidos, por ello es una de las etapas fundamentales del desarrollo de software.²³

El diseño de la arquitectura se da inmediatamente después de la especificación de los requerimientos del sistema y considera los siguientes elementos principales: Los componentes del software, las propiedades de dichos componentes y la comunicación entre ellos.

²³ <http://www.enterate.unam.mx/Articulos/2006/febrero/arquitect.htm> , abril 2015

Diseño en el nivel de componentes

De manera concreta, cuando se diseña una arquitectura de software, se crean y/o representan componentes que interactúen entre ellos y tengan una tarea específica asignada. Del mismo modo, estos componentes deben ser organizados para que logren los objetivos establecidos.

El desarrollo de software basado en componentes busca la reutilización de código previamente creado que es capaz de realizar diferentes tareas. Esto ha permitido beneficios tales como mejoras a la calidad de los productos y reducción en los ciclos de desarrollo.

Esencialmente, un *componente* es un fragmento de código escrito previamente que encapsula alguna funcionalidad expuesta a través de interfaces estándar²⁴, una vez que se combinan varios de ellos, pueden llevar a cabo una tarea específica.

El paradigma de ensamblar los componentes y escribir código que permita la comunicación entre ellos se conoce como *desarrollo de software basado en componentes*. Algunas ventajas de este paradigma son:

- Reutilización de software
- Simplificación de pruebas
- Simplificación del mantenimiento del sistema
- Mayor calidad

*Diseño de interfaz de usuarios*²⁵

Es importante contar con un adecuado diseño de la interfaz de usuario para que el sistema tenga éxito y su uso se facilite.

Se requiere del cumplimiento de algunos principios para el diseño de una buena interacción, como son:

- Buscar la consistencia. Debe utilizarse la misma terminología para avisos, menús y pantallas de ayuda, del mismo modo, deben usarse los mismos comandos en todo el proyecto
- Permitir que los usuarios frecuentes hagan uso de accesos directos. Cuando la frecuencia de uso de la aplicación aumenta, también aumenta el deseo del usuario por incrementar la velocidad de interacción; el uso de abreviaturas, teclas de función, etcétera, son muy útiles para los usuarios frecuentes

²⁴ WebCab Components, About Component Based Development, <http://webcabcomponents.com/componentization.shtml>. mayo 2015

²⁵ Schneiderman, Ben; Plaisant, Catherine. *Designing the user interface*. Ed. Pearson, 5a edición.

- Ofrecer retroalimentación informativa. Para cada acción del usuario, debe haber algún tipo de retroalimentación. Para las acciones más frecuentes y de menos importancia, la respuesta puede ser breve, mientras que para las acciones poco frecuentes y las más importantes, la respuesta debe ser más sustancial
- Diseñar acciones secuenciales. Las secuencias de acciones deben organizarse en grupos
- Errores simples. En la medida de lo posible, debe diseñarse un sistema que no permita que el usuario cometa un error irreparable. El sistema debe ser capaz de detectar el error y ofrecer mecanismos para recuperarse de él
- Reversión de las acciones sencillas. Permite a los usuarios “deshacer” acciones lo que fomenta la exploración de acciones desconocidas
- Apoyar al usuario con una interfaz amigable
- Reducir la carga de memoria

Implementación y prueba de unidades

En esta etapa del proceso, la creación del software se lleva a cabo como un conjunto de unidades o programas (en ella, se traduce lo planteado en la etapa anterior a una forma legible para la máquina, es decir, se genera código); la prueba de unidades implica validar que cada una de ellas cumpla con su especificación prevista.

Así pues, la escritura del código fuente es el principal esfuerzo de construcción del software, por ello es importante considerar los siguientes puntos:

- Aplicar técnicas para crear código comprensible (nombres de variables, formato de código, etc.)
- Manejar condiciones de error (errores previstos e imprevistos)
- Prevenir errores de seguridad en el código (llenado del búfer, introducción de datos erróneos, índices fuera de rango en matrices, etc.)
- Uso eficiente de recursos escasos (hilos, bloqueos en base de datos)
- Documentación de código

En esta fase es importante realizar verificaciones al sistema de manera modular, es decir, no se hacen pruebas del sistema completo sino de las partes específicas en que se está trabajando (la interfaz, el Web Service, la base de datos, etcétera), lo que ayuda a reducir errores de programación antes de las pruebas finales.

Integración y prueba del sistema

Una vez que se ha generado el código y se han realizado verificaciones a las unidades individuales, comienza la prueba del sistema completo para asegurar que se cumplen con los requerimientos establecidos en la primera etapa del desarrollo.

La prueba se centra en la lógica interna del software, y en las funciones externas, realizando diferentes ingresos de los datos para asegurar que los resultados son los esperados.

El propósito de las pruebas es reducir el tiempo de la detección de errores y garantizar la calidad del producto.

Algunas técnicas para poder realizar estas validaciones son:

- Realización de pruebas unitarias y de integración
- Escribir las pruebas (script de pruebas – test first development)
- Ejecución línea a línea (code stepping)
- Depuración (debugging)
- Revisiones
- Análisis estático

Después de realizadas las pruebas, el sistema se entrega al cliente.

Funcionamiento y mantenimiento

Ésta suele ser, por lo general, la etapa más larga del ciclo, en la cual el sistema está listo para funcionar en un ambiente real de acuerdo a lo previsto en la primera etapa. Sin embargo, es importante tomar en cuentas que el software sufrirá cambios después de que se entrega al cliente; el mantenimiento implica corregir errores no vistos en etapas anteriores, mejorar la implementación de las unidades del sistema y modificar los servicios cuando el cliente requiera ampliaciones funcionales o de rendimiento.

Existen diferentes tipos de mantenimientos según el momento en el que se localiza el error. Se enlistan a continuación:

Mantenimiento correctivo: Se lleva a cabo al detectarse los errores durante el funcionamiento

Mantenimiento perfectivo: Se lleva a cabo al decidirse ampliar o mejorar el software

Mantenimiento adaptativo: Se lleva a cabo al considerarse algún cambio en el software actual

Mantenimiento preventivo: Se realiza para adelantarse a posibles problemas

2.3 Teoría de JAVA

Java es un lenguaje orientado a objetos y de alto nivel que, si bien es cierto que su uso más destacado es en la Web, también permite escribir programas convencionales.²⁶

En Java, al igual que en otros lenguajes, el código fuente debe ser compilado y ejecutado, sin embargo en Java se buscó independencia de la máquina donde se ejecutarían los programas para darle mayor portabilidad. En la mayoría de los lenguajes, al compilar se genera un código máquina directamente ejecutable en una máquina determinada (y de uso limitado a ese tipo de máquina), en cambio, al compilar un programa desarrollado en Java, el código generado es independiente del hardware y no se puede ejecutar directamente en ninguna máquina, sino que ha de ser interpretado por una máquina virtual.

Es por ello que una de sus mayores ventajas es que sea independiente de la plataforma, esto significa que el código que genera el compilador puede ser transportado a cualquier plataforma con una máquina virtual de Java instalada y ser ejecutado.

De acuerdo a lo anterior, Java incluye un compilador y un intérprete. El primero genera el código en bytes que posteriormente será ejecutado por el intérprete Java conocido como máquina virtual Java (*JVM*), como puede verse en la Figura 10:



Figura 10. Flujo de los componentes del lenguaje de programación Java

Al hablar de código de bytes (*bytecode*), se hace referencia a un conjunto de instrucciones escritas en lenguaje máquina que no corresponden a ningún procesador en específico, sino a la máquina virtual de Java.

Así como se ha descrito a Java como un lenguaje independiente de la plataforma, también es importante mencionar algunas otras características notables, por ejemplo, es robusto, es capaz de gestionar la memoria automáticamente, no permite el uso de técnicas de programación inadecuadas, permite la programación multithread, permite crear aplicaciones cliente-servidor, contiene mecanismos de seguridad propios, además de que posee herramientas de documentación incorporadas.

²⁶ Ceballos, Francisco Javier. *Java 2 Curso de programación*. Ed. Alfaomega, 4a edición. México 2011, p. 7-8

Por otro lado, es importante destacar que al programar en Java no se parte desde cero pues cualquier aplicación desarrollada en él está basada en clases ya existentes creadas, ya sea por el propio usuario, o de uso comercial.

2.3.1 Historia de JAVA

Java fue desarrollado por *Sun Microsystems* en 1991. Inicialmente fue creado para comunicar aparatos electrónicos de consumo como videos, televisiones, equipos de músicos, etcétera, sin embargo la escasa capacidad de memoria y la diversidad de procesadores dificultó el proyecto, provocando que se buscara un lenguaje de programación independiente del dispositivo el cual fue conocido inicialmente como *Oak*.

Con el nacimiento de Internet en 1994, el grupo de investigación de Sun se dio cuenta que los avances de su investigación eran aplicables a él y cambiaron el rumbo de su proyecto hacia el desarrollo de un lenguaje que permitiera crear aplicaciones que pudieran ser ejecutadas desde cualquier ordenador conectado a Internet, siendo necesario únicamente el navegador.

A partir de ello, se empezó a conocer más de Java y de sus aplicaciones, conocidas como applets. Un applet es un programa escrito en Java ejecutado desde una página Web en cualquier PC, independiente del sistema operativo que tenga y la arquitectura de su procesador.

En enero de 1995, *Sun* formó la empresa *Java Soft* dedicada al desarrollo de productos basados en tecnologías Java y para crear aplicaciones, herramientas, sistemas de plataforma y servicios para aumentar las capacidades del lenguaje.²⁷

Java es un lenguaje muy completo, que se está convirtiendo rápidamente en un macrolenguaje (*Java 1.0* tenía 12 packag; *Java 1.1* tenía 23 y *Java 1.2* tiene 59), *Sun* lo describe como “*simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico*”. Aunque en algunas características Java tenga mucho que mejorar, esta descripción lo detalla bastante bien.

Puede decirse que *Java 1.2* (o *Java 2*) no cuenta con diferencias importantes con respecto a *Java 1.1* (la cual sí dista mucho de su antecesora *Java 1.0*), sino que son extensiones y ampliaciones.

En la actualidad, la tecnología Java puede encontrarse en redes y dispositivos que van desde Internet y supercomputadoras científicas hasta laptops y teléfonos móviles.

²⁷ http://www.cad.com.mx/historia_del_lenguaje_java.htm, Marzo 2015

Ceballos, Francisco Javier. *Java 2 Curso de programación*. Ed. Alfaomega, 4a edición. México 2011, p. 8

Otra de las características de la historia de este lenguaje es el logo (Figura 11). Se describe como una taza de café humeante, se dice que Sun Microsystems primero bautizó a su software sin saber que Java es una isla de Indonesia ligeramente famosa por su producción de café.



Figura 11. Logo JAVA²⁸

2.3.2 Características de la programación orientada a objetos

Los antecedentes de lenguaje de programación JAVA está fundada en el lenguaje de programación C y C++, éste se caracteriza por ser poco amigable para el desarrollador. La evolución de los lenguajes de programación ha sido muy radical según las necesidades de las tecnologías desde cómo comunicar una base de datos o comunicar dispositivos móviles. Es por ello importante mencionar brevemente cómo ha sido esta evolución. Se muestra en la Figura 12.

²⁸ ORACLE www.oracle.com/java/index.html , Abril 2015

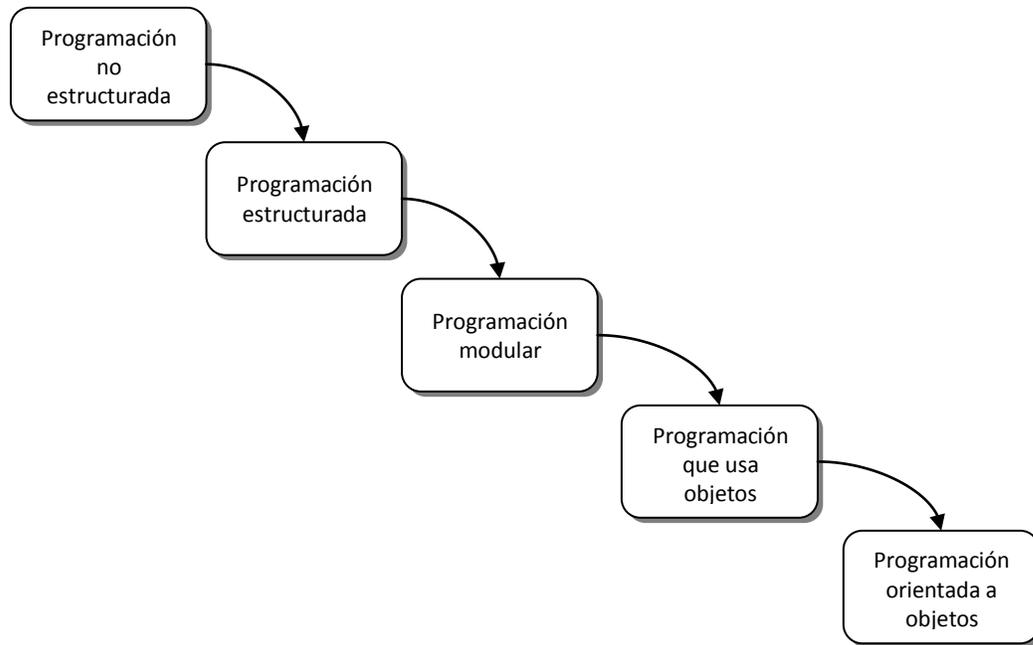


Figura 12. Evolución de los lenguajes de programación

La programación no estructurada se caracteriza por un flujo de diagramas de difícil seguimiento y un uso indiscriminado del comando GOTO. Los lenguajes característicos son Fortran, Basic y primeras versiones de COBOL.

La programación estructurada se caracteriza por una ordenación del flujo de programas y una restricción del uso del comando GOTO. Los principales lenguajes PASCAL, ADA y C.

La programación modular se caracteriza por la ordenación del código en módulos, funciones y procedimientos en los años 80s. Aceptada por la mayoría de los lenguajes.

La programación que usa objetos se caracteriza por la existencia de objetos con propiedades y métodos disponibles para el programador en los años 90s. Ejemplo Visual Basic.

La programación orientada a objetos se caracteriza por los programas que conciben un conjunto de objetos como representación del problema real. Esta característica la poseen numerosos lenguajes como Visual Basic .NET, Java. A continuación se detalla la programación orientada a objetos.

La programación orientada a objetos (POO) es un paradigma de programación que utiliza objetos ligados a través de mensajes. Está basado en varias técnicas como herencia, abstracción, polimorfismo, encapsulamiento, etcétera.

La orientación a objetos se enfoca en generar mejoras en la forma de diseño, desarrollo y mantenimiento del software ofreciendo soluciones a largo plazo a los problemas que han siempre existido en el desarrollo de software:

- Falta de portabilidad del código y escasa reutilización
- Código difícil de modificar
- Ciclos de desarrollo largos

En ese sentido, un lenguaje orientado a objetos tiene tres características fundamentales: debe estar basado en objetos y clases, y debe ser capaz de heredar entre clases.

Dado que la POO depende de la existencia de un mundo lleno de objetos y que la resolución de cualquier problema se realiza en términos de objetos, un lenguaje se dice que está basado en objetos si soporta a los mismos como una característica esencial del mismo. Conviene aclarar la diferencia entre estar basado en y ser orientado a objetos pues, para que sea orientado necesita tener clases y relaciones de herencia entre ellas.

*Objetos*²⁹

El elemento más importante de la POO es el objeto, el cual puede definirse como un conjunto complejo de datos y programas que poseen estructuras y forman parte de una organización. Es decir, un objeto no es un dato simple pues contiene cierto número de componentes bien estructurados y, además, no es un ente aislado sino que forma parte de una organización jerárquica o de otro tipo.

La combinación de los objetos es un aspecto de este tipo de programación, pero incluye también aspectos y características que hacen que la creación y el uso de objetos sea sencillo y flexible. Uno de los mecanismos más importantes es la *clase*, por otro lado tenemos a la herencia y al encapsulamiento como dos de las principales propiedades.

Por encapsulamiento, debemos entender que un objeto es auto-contenido, es decir, una clase puede definirse como la generalización de un tipo específico de objetos con la misma estructura (atributos) y el mismo comportamiento (métodos), pero cada uno tendrá sus propias características.

Cuando se utiliza POO se definen clases (las cuales definen objetos genéricos) y la forma en que los objetos van a interactuar entre ellos a través de mensajes. Al crear un objeto de una clase determinada, se dice que se crea una *instancia* de la clase (o un objeto propiamente dicho). La ventaja de esto radica en que no hay programas que modifiquen al

²⁹ Ceballos, Francisco Javier. *Java 2 Curso de programación*. Ed. Alfaomega, 4a edición. México 2011, p. 12

objeto, éste se mantiene hasta cierto punto independiente, si es necesario modificarlo (agregarle características), se puede hacer sin tocar el resto de la aplicación lo que ahorra tiempo.

Otros conceptos fundamentales para este paradigma de programación son el encapsulamiento (en referencia a qué atributos y métodos forman el objeto como una unidad única) y la ocultación de información (en cuanto que los atributos deben estar lo más ocultos posibles al exterior y ser manipulados o consultados a través de los métodos permitidos y pertinentes). Ambos conceptos son representados en la Figura 13.

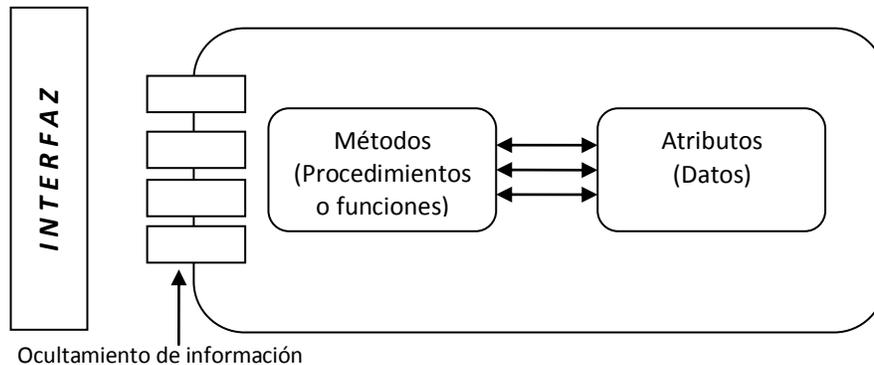


Figura 13. Encapsulamiento

Por otro lado, la herencia se refiere a que pueden crearse nuevas clases que hereden características (atributos y métodos) de otras ya existentes; esto simplifica la programación porque las clases hijas incorporan automáticamente los métodos de las madres garantizando la reutilización del código.

Con la herencia cada clase se clasifica en una jerarquía estricta y cada una tiene su superclase (la clase superior en la jerarquía llamada también clase base), asimismo pueden tener una o varias subclasses (las clases inferiores en la jerarquía llamadas también clases derivadas).

Clases

Anteriormente se ha hablado de objetos y clases pero no se ha definido lo que son éstas últimas, aun cuando el concepto más adecuado para definir las características de los objetos de un sistema es el propio concepto de clase.

Una clase es la descripción de un conjunto de objetos con una estructura igual (atributos) y el mismo comportamiento (métodos). Así, puede decirse que las clases son el patrón que proporcionan los lenguajes orientados a objetos para definir la implementación de los objetos.

Todos los objetos son instancias de una clase, es decir, ejemplos concretos de una clase. Como puede verse, la clase lo define todo, es una entidad autocontenida en cuanto que contiene todo lo necesario para definir la estructura o los datos (atributos) que cada objeto tendrá y la manera de manejarse esos objetos a través de los métodos.

Es decir, cada clase posee un doble componente:

- Componente estático, los datos, que son campos con nombres que poseen ciertos valores. Estos campos caracterizan el estado del objeto (atributos).
- Componente dinámico, los procedimientos o funciones (métodos) que representan el comportamiento común de los objetos que pertenecen a una misma clase. Son los métodos los que se encargan de modificar los campos de los objetos (manipular los atributos de forma directa rompería el encapsulamiento y la ocultación de la información) y caracterizan las acciones que pueden realizarse sobre los mismos.

Es conveniente aclarar que no todos los métodos son accesibles para los usuarios, puede que sólo sean visibles para un grupo concreto de clases o únicamente por la clase en que están siendo definidos.

2.3.3 Características del lenguaje

Entre algunas de las características más importantes se encuentra la robustez, pues debido a la forma en que está diseñado, Java no permite el manejo directo de hardware ni de memoria. El compilador, por otro lado, no permite utilizar variables sin inicializar, no permite modificar valores de punteros directamente o acceder a métodos y variables de forma incorrecta, tampoco permite la herencia múltiple, etcétera.

Java implementa, además, mecanismos de seguridad que limitan el acceso a recursos de las máquinas donde está siendo ejecutado, especialmente en el caso de los applets. De la misma manera, está diseñado para trabajar en la red, de modo que incorpora objetos que permiten acceder a archivos de manera remota.

Adicionalmente, con el JDK (Java Development Kit) vienen incorporadas varias herramientas, tales como un generador automático de documentación.

Sun describe el lenguaje Java como se muestra en la Figura 14:

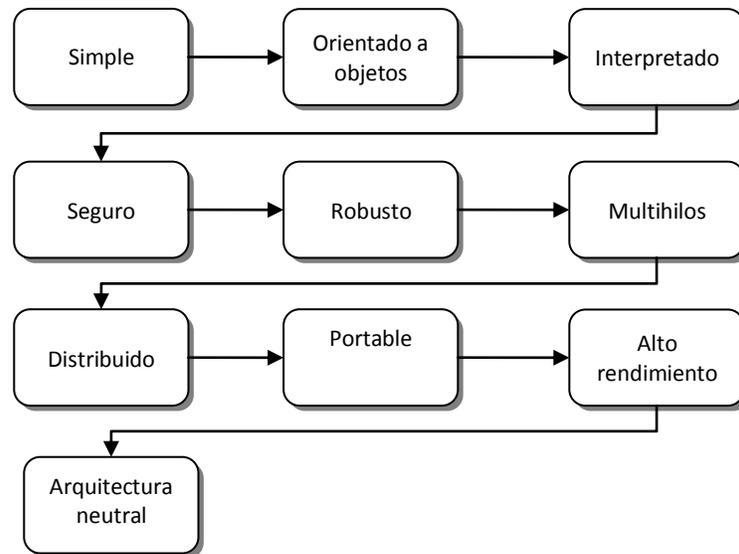


Figura 14. Características de lenguaje de programación JAVA

- **Simple**
Se le conoce como lenguaje simple porque viene de la misma estructura de C y C++, por lo cual utiliza determinadas características de ellos.
- **Orientado a Objetos**
En su mayoría, la programación en Java está orientada a objeto, ya que al estar agrupados en estructuras encapsuladas es más fácil su manipulación.
- **Interpretado**
Java puede ser compilado e interpretado en tiempo real, ya que cuando se construye el código fuente éste se transforma en una especie de código de máquina.
- **Seguro**
Se han implementado barreras de seguridad en el lenguaje y en el sistema de ejecución de tiempo real.
- **Robusto**
Es altamente fiable en comparación con C, se han eliminado muchas características con la aritmética de punteros y proporciona numerosas comprobaciones en compilación y en tiempo de ejecución.
- **Multihilos**
Java es muy dinámico en la fase de enlazado, sus clases solamente actuarán en la medida en que sean requeridas o necesitadas. Con esto, permitirá que los enlaces se puedan incluir desde fuentes muy variadas o desde la red.

- **Distribuido**

Permite abrir sockets, establecer y aceptar conexiones con los servidores o clientes remotos; facilita la creación de aplicaciones distribuidas ya que proporciona una colección de clases para aplicaciones en red.

- **Portable**

Por ser indiferente a la arquitectura sobre la cual está trabajando, su portabilidad es muy eficiente; sus programas son iguales en cualquiera de las plataformas ya que Java especifica tamaños básicos, esto se conoce como la máquina virtual de Java.

- **Alto rendimiento**

Java es considerado de alto rendimiento por ser tan veloz al momento de correr los programas y por ahorrarse muchas líneas de código.

- **Arquitectura neutral**

Java es compatible con los más variados entornos de red, desde Windows 95 a Windows NT, Unix y Mac, con la finalidad de poder trabajar con diferentes sistemas operativos.

2.4 Web Service

A medida que las tecnologías avanzan también aumenta su demanda por ello, en un círculo de progreso, al existir una mayor demanda existe la necesidad de crear nuevas tecnologías para poder soportar el crecimiento que conlleva una población en aprendizaje.

Al inicio las computadoras solían ejecutar un único programa a la vez, pero en la medida que se requería correr más de un programa simultáneamente, surgió la necesidad de contar con los mecanismos de comunicación entre ellos, este mecanismo ha evolucionado debido que los computadoras conforman redes. Por tanto, esta comunicación debe poder efectuarse entre un programa, que corre en la computadora principal, y otro programa, que corre en la otra.

Para que esta comunicación funcione, primero debe existir un medio de comunicación entre las computadoras; esto hoy está resuelto con la Internet. Y segundo, el programa debe saber conversar con cualquier plataforma. Para que esto ocurra el programador a cargo de los equipos debe conocer ambos.

La solución que se propone con el uso del Web Service es que no hay necesidad que el programador conozcan los dos equipos o todos los programas, de modo que, quien quiera recibir los datos, sólo necesita usar el Web Service.

Esto significa que pueden existir transferencias de datos entre distintas aplicaciones –programas- que funcionan en varias computadoras, con distintos sistemas operativos y que pertenezcan a diferentes empresas o instituciones, un mundo de comunicación (Figura 15).



Figura 15. Importancia de Web Service

2.4.1 Concepto de Web Service

El Web Service es una aplicación en clientes y servidores que se comunican a través de HyperText Transfer Protocol (HTTP) de World Wide Web (WWW). De acuerdo al Consorcio World Wide Web (W3C), los Web Service proporcionan un estándar de interoperabilidad entre aplicaciones de software que son ejecutados en diferentes plataformas y frameworks.³⁰

El Web Service se caracterizan por su gran interoperabilidad y extensibilidad así como por su facilidad de procesamiento, gracias al uso de XML (eXtensible Markup Language). Otra ventaja importante que presentan es que pueden ser combinados libremente para lograr operaciones complejas. Las principales características que hacen al Web Service como una tecnología viable se mencionan a continuación:

- Los servicios web son componentes de la aplicación
- Los servicios web se comunican utilizando protocolos abiertos
- Los servicios web son autónomos y auto-descripción

³⁰ <http://docs.oracle.com/javaee/6/tutorial/doc/gijti.html>, Marzo 2015

- Los servicios web pueden ser descubiertos mediante UDDI
- Los servicios web pueden ser utilizados por otras aplicaciones
- HTTP y XML es la base para servicios web

Por otro lado, se basan en un conjunto de estándares de comunicación, como son XML para la representación de datos, SOAP (*Simple Object Access Protocol*) para el intercambio de datos y el lenguaje WSDL (*Web Service Description Language*) para describir las funcionalidades del servicio.

También existen distintas funcionalidades para el descubrimiento de Web Service, gestión de eventos, archivos adjuntos, seguridad, gestión y fiabilidad en el intercambio de mensajes y transacciones.

Sin embargo sus principales usos se enfocan a reutilizar aplicaciones y componentes. Además de poder ayudar a resolver el problema de la interoperabilidad, dando distintas aplicaciones una forma de vincular sus datos.

2.4.2 Estructura de Web Service

A nivel conceptual, un servicio es un componente de software proporcionado a través de un *endpoint* accesible en la red. El proveedor y el consumidor del servicio utilizan mensajes para intercambiar peticiones de invocación y respuestas como documentos autocontenidos.

A nivel técnico, el Web Service puede ser implementado de varias formas.

A continuación se describen dos de ellas son los Big Web Services y RESTful Web Service.

Big Web Service

El Big Web Service utiliza los mensajes XML que siguen el protocolo de acceso simple de objetos (SOAP), un lenguaje XML que define la arquitectura y el formato de un mensaje. Estos sistemas a menudo contienen una descripción de las operaciones legible para máquina que ofrece el servicio, escritas en el Lenguaje de Descripción de Web Service (WSDL), un lenguaje XML para definir interfaces sintácticamente.

RESTful Web Service

Para Java EE6, JAX-RS provee una funcionalidad para Web Service de transferencia de estado representacional (RESTful). REST es útil para escenarios básicos y con integración ad-hoc. Una característica importante de este tipo de Web Service es que se integran mejor con HTTP que los servicios basados en SOAP, debido a que no requieren mensajes XML o definiciones WSDL.

Dado que los RESTful Web Service utiliza estándares bien conocidos de W3C y de Internet Engineering Task Force (IETF) (tales como HTTP, XML, URI, MIME) y tienen una infraestructura ligera que permite a los servicios construir con un uso mínimo de herramientas, el desarrollo de servicios RESTful es barato de realizar.

2.4.3 Arquitectura SOA

La Arquitectura SOA (Service Oriented Architecture) establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante servicios web, una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular (Tabla 7).

Tabla 7. Capacidades de una arquitectura de referencia SOA

Servicios	Descripción
Interacción con Usuarios	Una arquitectura de referencia SOA requiere la capacidad de interacción de usuarios finales con los servicios tales a través de portales personalizables y herramientas web basadas en roles, lo cual mejora la productividad y efectividad de las soluciones
Orquestación de Servicios	Una arquitectura de referencia SOA requiere la capacidad de que los servicios puedan ser ensamblados en un proceso completo de negocios utilizando tecnologías para orquestación
Administración de Servicios	Una arquitectura de referencia SOA requiere la capacidad de administración de servicios, el cual permita monitorear qué están realizando los servicios y provea funcionalidades de gobierno de los servicios
Registro de Servicios	Arquitectura de referencia SOA requiere la capacidad de un registro centralizado a través del cual los servicios pueden ser catalogados en un repositorio central
Comunicaciones y Mensajería	La manera más flexible de unir los servicios es optar por una arquitectura bajamente acoplada, independiente de la plataforma, basada en principios de mensajería para comunicación entre componentes
Servicios de Negocios	Servicios de negocios reutilizables. Capacidad para creación y exposición de servicio a partir de componentes o programas existentes o creación de nuevos servicios

2.4.4 Protocolo SOAP

Es un protocolo que permite la comunicación entre aplicaciones a través de mensajes por medio de Internet. Es independiente de la plataforma, y del lenguaje. Está establecido en XML y es la base principal del Web Service. Los mensajes SOAP son documento XML propiamente dicho.

Este protocolo basado en XML consiste de tres partes: Un sobre (envelope), el cual define qué hay en el mensaje y cómo procesarlo; un conjunto de reglas de codificación para expresar instancias de tipos de datos; y una convención para representar llamadas a procedimientos y respuestas.

Un ejemplo de un mensaje SOAP en donde se observa las tres partes del protocolo.

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  Soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Header>
  ...
  </soap:Header>

  <soap:Body>
  ...
  <soap:Fault>
  ...
  </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

El protocolo SOAP tiene tres características principales (Figura 16):

- Extensibilidad (seguridad y WS-routing son extensiones aplicadas en el desarrollo)
- Neutralidad (SOAP puede ser utilizado sobre cualquier protocolo de transporte como HTTP, SMTP, TCP o JMS)
- Independencia (SOAP permite cualquier modelo de programación)

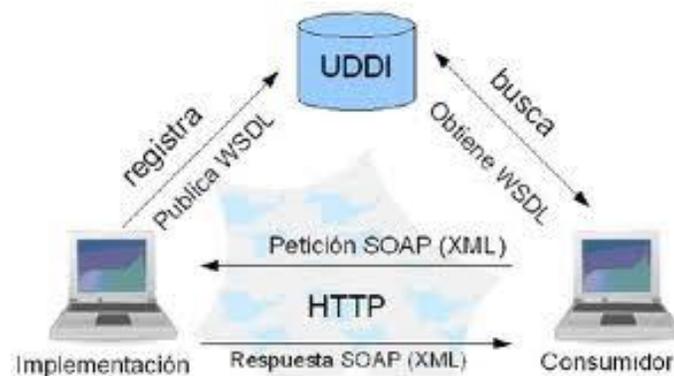


Figura 16. Arquitectura del protocolo SOAP

2.5 Lenguaje .NET

El lenguaje .NET es un proyecto de Microsoft utilizado para crear una plataforma de desarrollo de software con transparencia de redes, con independencia de plataforma y en donde se permita el rápido acceso a las aplicaciones.

Surge como respuesta al creciente mercado de los negocios en entornos web y como competencia de Java Sun Microsystems.

Con esta plataforma, Microsoft incursiona de lleno en el campo de los Servicios Web y establece el lenguaje XML como norma en el transporte de información en sus productos y lo promociona como tal en los sistemas desarrollados utilizando sus herramientas.

.NET intenta ofrecer una manera rápida y económica pero a la vez segura y robusta de desarrollar aplicaciones –o como la misma plataforma las denomina, soluciones– permitiendo a su vez una integración más rápida y ágil entre empresas y un acceso más sencillo y universal a todo tipo de información desde cualquier tipo de dispositivo.

Los framework constituyen la base de la plataforma .NET y muestran la infraestructura sobre la cual se reúnen un conjunto de lenguajes, herramientas y servicios que simplifican el desarrollo de aplicaciones en entorno de ejecución distribuido.

2.5.1 Características de .NET

A continuación se enlistan las principales características del lenguaje de programación de .NET:

- Las aplicaciones .NET se ejecutan en un entorno específico del sistema operativo denominado *runtime*, lo que lo hace flexible, seguro y portable
- Es orientado a objetos y permite desarrollar aplicaciones en más de un lenguaje de programación
- Está diseñado para lograr el desarrollo de aplicaciones complejas, robustas y flexibles
- Proporciona un único modelo de programación consistente para el desarrollo de diferentes modelos de aplicaciones como Windows, la web, consolas o móviles o bien, para diferentes dispositivos de hardware como PC, tablets, entre otros
- Puede integrarse fácilmente con las aplicaciones desarrolladas en versiones anteriores
- Se puede integrar con otras plataformas y sistemas operativos al implementar estándares como XML, SOAP, WSDL, etc.

2.5.2 Componentes del lenguaje .NET

Los principales componentes del framework (Figura 17) son:

- El conjunto de lenguajes de programación
- La Biblioteca de Clases Base (BCL, por sus siglas en inglés)
- El Entorno Común de Ejecución para Lenguajes o CLR por sus siglas en inglés

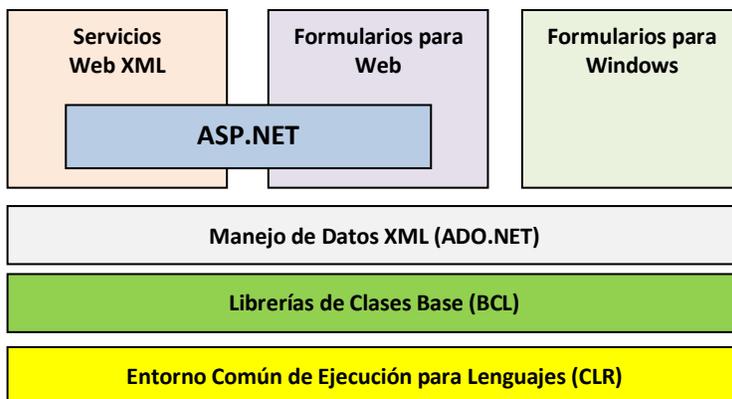


Figura 17. Componentes del Framework .NET

Debido a la publicación de la norma para la CLR, el desarrollo de lenguajes se facilita, por lo que el marco del framework .NET soporta ya más de 20 lenguajes de programación y es posible desarrollar cualquiera de los tipos de aplicaciones soportados en la plataforma con cualquiera de ellos, lo que elimina las diferencias que existían entre lo que era posible hacer con uno u otro lenguaje.

Algunos de los lenguajes desarrollados para el framework .NET son: C#, Visual Basic, Delphi (Object Pascal), C++, J#, Perl, Python, Fortran y Cobol.NET.

El CLR es el verdadero núcleo del Framework de .NET, es el entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando con ello el conjunto de servicios del sistema operativo.

La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .NET en un código intermedio. Para ejecutarse se necesita un segundo paso, un compilador JIT (Just-In-Time) es el que genera el código máquina real que se ejecuta en la plataforma del cliente. De esta forma se consigue, con .NET, independencia de la plataforma hardware.

La compilación JIT la realiza el CLR a medida que el programa invoca métodos, el código ejecutable obtenido, se almacena en la memoria caché del ordenador, siendo

recompilado de nuevo sólo en el caso de producirse algún cambio en el código fuente (lo que implicaría volver a compilar el código).

La BCL maneja la mayoría de las operaciones básicas que se encuentran involucradas en el desarrollo de aplicaciones incluyendo, entre otras:

- Manejo de datos
- Administración de memoria
- Cifrado de datos
- Transmisión y recepción de datos por distintos medios (XML, TCP/IP)
- Administración de componentes Web que corren en el servidor como en el cliente
- Herramientas de seguridad e integración con la seguridad del sistema operativo
- Interacción con otras aplicaciones
- Manejo de cadenas de caracteres y expresiones regulares
- Operaciones aritméticas
- Manipulación de fechas, zonas horarias y periodos de tiempo
- Manejo de arreglos de datos y colecciones
- Manipulación de archivos de imágenes
- Aleatoriedad
- Generación de código
- Windows API
- Compilación de código

Esta funcionalidad se encuentra organizada por medio de espacios de nombres jerárquicos.

La biblioteca de clases base se clasifica, en tres grupos clave:

- ASP.NET y Servicios Web XML
- Windows Forms
- ADO.NET

Los ensamblados son ficheros con forma de EXE (EXEcutable) o DLL (DyNamic-Link Library) que contienen toda la funcionalidad de la aplicación de forma encapsulada. Gracias a ellos, ya no es necesario registrar los componentes de la aplicación.

En la Figura 18 se muestra la relación de Common Language Runtime y la biblioteca de clases con las aplicaciones y el sistema en su conjunto. En la ilustración se representa igualmente cómo funciona el código administrado dentro de una arquitectura mayor.

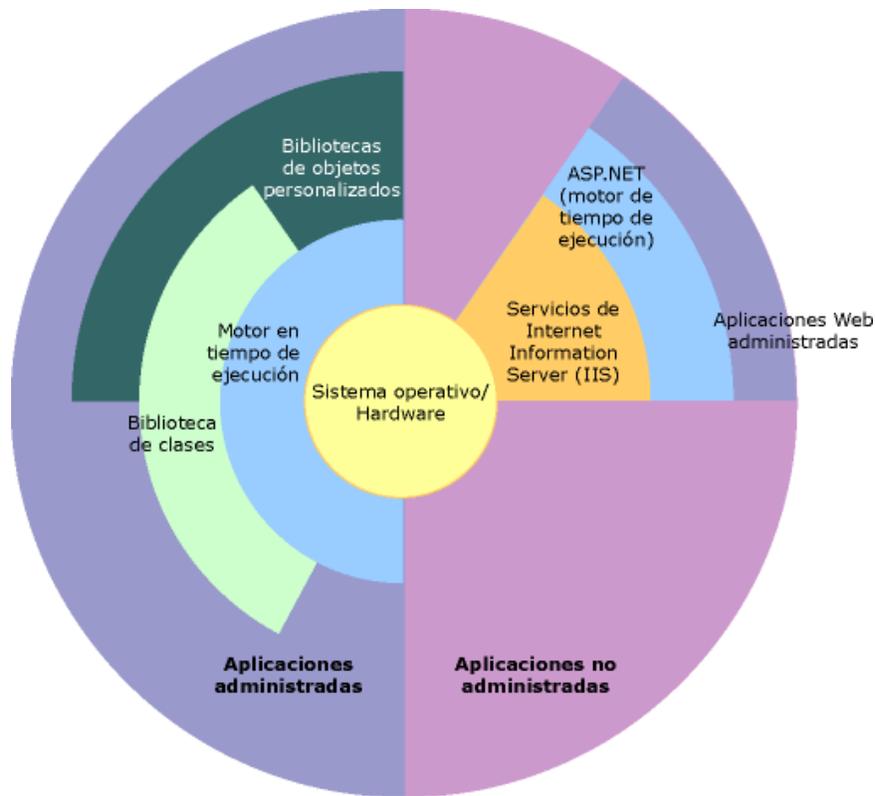


Figura 18. .NET Framework en contexto

2.5.2.1 Características de Common Language Runtime (CLR)

CLR administra la memoria del sistema, ejecución de subprocessos, ejecución de código, comprobación de la seguridad del código, compilación y demás servicios del sistema. Estas características son intrínsecas del código administrado que se ejecuta.

El CLR impone la seguridad de acceso del código. Sus características de seguridad permiten que el software legítimo implementado en Internet sea excepcionalmente variado, además de imponer la solidez del código mediante la implementación de una infraestructura capaz de comprobación de tipos y código denominado CTS (Common Type System, Sistema de Tipo Común). CTS garantiza que todo el código administrado es autodescriptivo.

Adicionalmente, elimina problemas del software común, como liberar objetos y administrar referencias cuando éstos ya no son usados. Esta administración automática de la memoria soluciona los dos errores más comunes de las aplicaciones: La pérdida de memoria y las referencias no válidas a la misma.

Además, el motor en tiempo de ejecución aumenta la productividad del programador.

Aunque el CLR está diseñado para el software futuro, también es compatible con el software actual y el software antiguo. La interoperabilidad entre el código administrado y no administrado permite que los desarrolladores continúen utilizando los componentes COM y las DLL que necesiten (Figura 19).

Por otro lado, el runtime se puede hospedar en aplicaciones de servidor de gran rendimiento, como Microsoft SQL Server e *Internet Information Services* (IIS).

Esta infraestructura permite utilizar código administrado para escribir lógica empresarial, al tiempo que se disfruta del superior rendimiento de los mejores servidores empresariales del sector que puedan hospedar el motor en tiempo de ejecución.

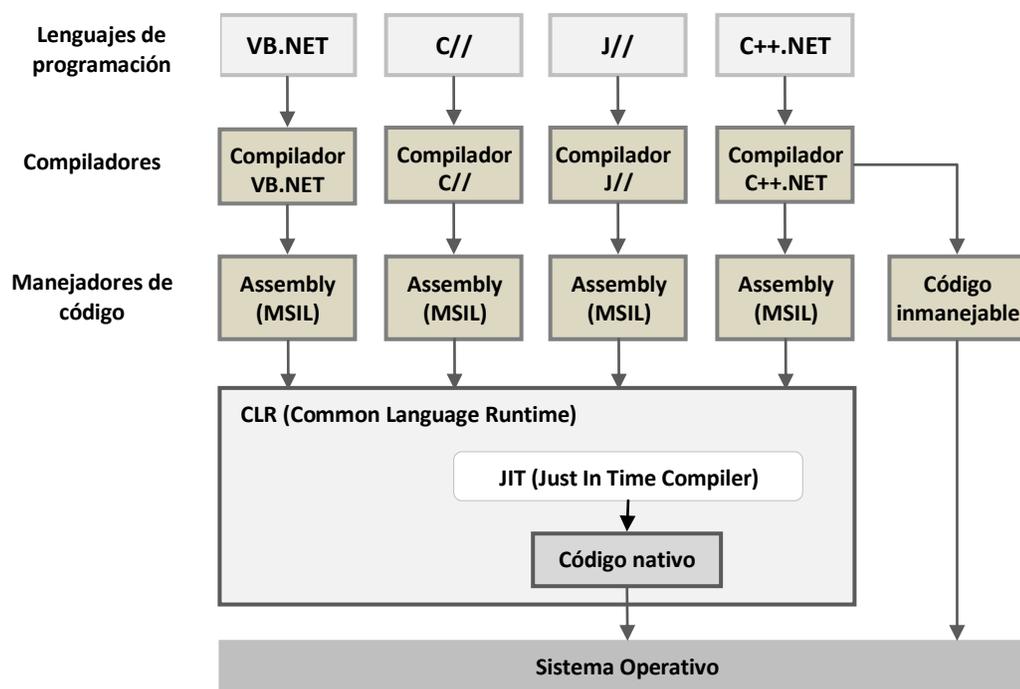


Figura 19. Ejecución CLR ³¹

2.5.2.2 Biblioteca de clases de .NET Framework

Esta biblioteca es una colección de tipos reutilizables que se integran estrechamente con CLR y la programación orientada a objetos, lo que proporciona tipos en los cuales su propio código administrado puede derivar funciones, esto ocasiona que sea más sencillo el uso del lenguaje reduciendo a la vez el tiempo de aprendizaje.

Por ejemplo, las clases de colección de .NET Framework implementan un conjunto de interfaces que puede usar para desarrollar sus propias clases de colección.

³¹ <http://img.redusers.com/imagenes/libros/lpcu143>, Febrero 2015

Permite realizar diversas tareas de programación comunes, como son la administración de cadenas, recolección de datos, conectividad de bases de datos y acceso a archivos. Además de estas tareas habituales, la biblioteca de clases incluye tipos adecuados para diversos escenarios de desarrollo especializados.

Se pueden realizar diversos tipos de aplicaciones y servicios como los siguientes:

- Aplicaciones de consola
- Aplicaciones GUI (Graphical User Interface) de Windows (Windows Forms)
- Aplicaciones de ASP.NET
- Servicios de Windows
- Aplicaciones orientadas a servicios utilizando Windows Communication Foundation
- Aplicaciones habilitadas para el flujo de trabajo utilizando Windows Workflow Foundation (WF)

Por ejemplo, las clases de Windows Forms son un conjunto completo de tipos reutilizables que simplifican enormemente el desarrollo de interfaces GUI (Graphical User Interface) para Windows. Si escribe una aplicación Web Form de ASP.NET, puede utilizar las clases de formularios Web Forms³². La Figura 20 muestra algunas de las clases de las librerías principales utilizadas en .NET:

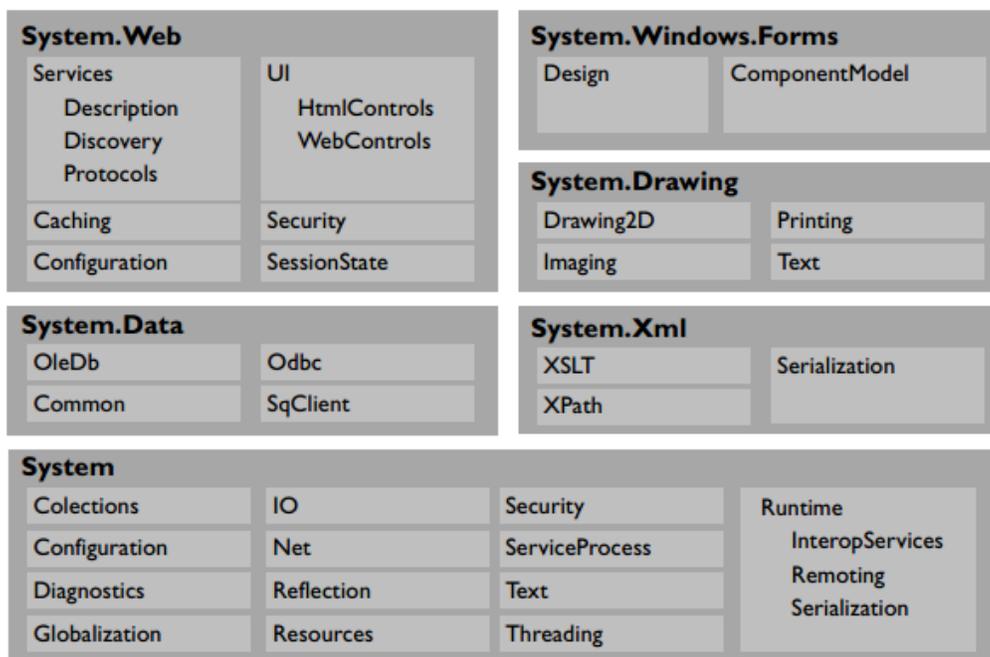


Figura 20. Principales Clases de las librerías NET Framework³³

³² [http://msdn.microsoft.com/es-es/library/zw4w595w\(v=vs.110\).aspx](http://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.110).aspx), Febrero 2015

³³ <http://img.redusers.com/imagenes/libros/lpcu143/capitulogratis.pdf>, Enero 2015

2.5.3 Entorno integrado de desarrollo con Visual Studio

Entorno realizado para aprovechar todas las características del Framework .NET, dentro de esta IDE (Integrated Development Environment) podemos realizar todas las aplicaciones Windows y web, servicios web y bibliotecas de componentes, además de acceder a una base de datos.

El IDE proporciona herramientas tales como compiladores, un depurador y otras destinadas a la administración del proyecto, está compuesto por cinco secciones (Figura 21):

- Barra de herramientas
- Lista de errores
- Cuadro de herramientas
- Explorador de soluciones y ventana de propiedades
- Área de trabajo

Debido a que el entorno se caracteriza por ser amigable para el usuario, será éste el entorno utilizado en el presente trabajo como base para el desarrollo de la interfaz de usuario.

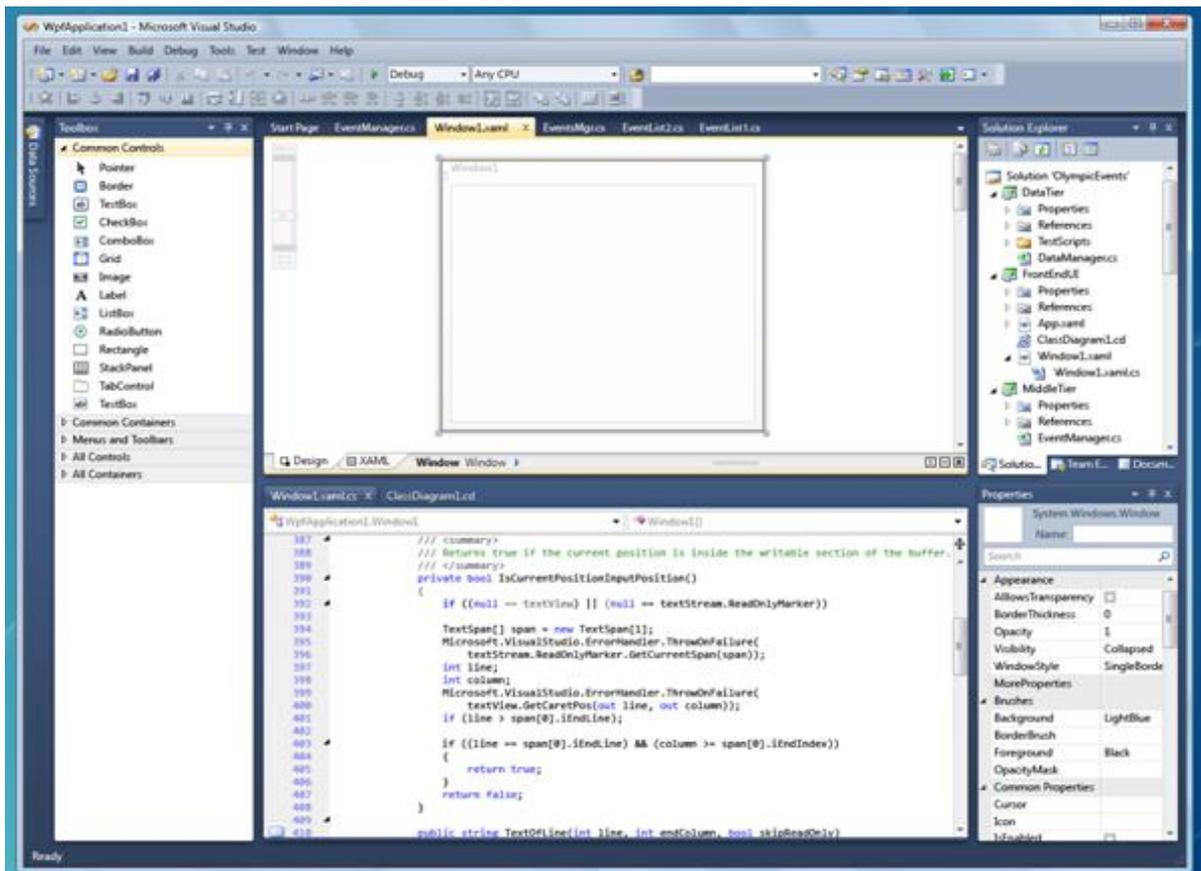


Figura 21. IDE Visual Studio

2.5.4 Entorno bancario y base de datos

Un sistema bancario debe proveer una poderosa flexibilidad y funcionalidad debido a la escalable y avanzada arquitectura que posee. Además debe permitir bajos costos de mantenimiento y usar estándares establecidos como XML, J2EE, HTTP entre otros para facilitar su crecimiento.

Algunas otras características que debe poseer el entorno son:

Eficiencia empresarial: Debe tener flexibilidad para satisfacer las necesidades de cualquier banco de la operación totalmente nueva más pequeña hasta la más grande multinacional. Esto ayuda a reducir el costo de propiedad, mejorar la productividad, aumentar la eficiencia y la mitigación de riesgos, ampliar los canales de comercialización, reducir los costos, mejorar el servicio al cliente, facilitar la venta y oportunidades de venta cruzada; además de ayudar a impulsar el crecimiento y aumentar la rentabilidad.

Despliegue en sectores múltiples: Proporcionar una base de alto rendimiento para gestionar usuarios, clientes, procesos, riesgos, canales y aplicaciones de administración, así como desplegar una gama completa de servicios bancarios a través de líneas especializadas de negocio. Dar cobertura a través de: minorista, banca corporativa, al por mayor, universal, gestión de patrimonio privado y microfinanzas.

Manejar grandes volúmenes, poseer alta seguridad y escalabilidad: Llevar volúmenes altos basados en múltiples, seguras y escalables servidores, no limita el número de transacciones, los usuarios o clientes. Esto asegura que es capaz de soportar cualquier tamaño de la institución financiera y todos los niveles de tráfico.

Estándares abiertos: Codificadas para la fiabilidad y de alto rendimiento proporcionados en lenguaje abierto C o lenguajes Java, basados en estándares de la industria establecidos, independientes y no propietarios. Tener hardware y bases de datos, que utilizan un servidor de aplicaciones J2EE abierta, proporcionando una conectividad abierta a través de XML y Web Service y que cuenta con una interfaz de usuario basada en el navegador abierto con HTML y XSLT. Único software de banca central capaz de ejecutar de forma nativa tanto en entornos Java.NET, lo cual significa que el usuario pueda elegir el mejor proveedor y/o ambiente para satisfacer sus propias necesidades, y realizar cambios sin alterar su inversión para una máxima longevidad de la inversión. Soporte nativo significa que puede utilizarse Java local o .NET de programación para adaptar la funcionalidad y flexibilidad del entorno. Esto amplía la funcionalidad y flexibilidad local sin comprometer la capacidad de actualizar a versiones posteriores.

2.5.5 Tecnología

Dentro del sector financiero es necesario cubrir ciertos requerimientos básicos que son solicitados por normas mexicanas, esto dependiendo del tamaño y la clasificación de la financiera, como la seguridad en las redes y en las aplicaciones que han tenido un gran peso en la actualidad.

Las bases de datos funcionan como una herramienta fundamental para poder llevar la administración de un sistema financiero no importando el tamaño de ésta ya que todos buscan la integridad de su información debido a la sensibilidad.

Por ello se hará mención de algunas de las características que debe tener una base de datos haciendo énfasis en su importancia.

2.5.6 Base de datos

Las bases de datos se han constituido como una de las herramientas mayormente difundidas en la actual sociedad de la información.

Una base de datos es un almacén que permite guardar grandes cantidades de información de manera organizada para poder administrarla de manera más eficiente y poder modificar y procesar la información más rápidamente cuidando su integridad.

Las bases de datos permiten almacenar información de manera segura y contando con una disponibilidad de la información no importando el lugar ni la ubicación del requerimiento (Figura 22).



Figura 22. Bases de datos y su importancia

2.5.6.1 Historia de las bases de datos

Las bases de datos se desarrollan a partir de la necesidad de las industrias de almacenar grandes cantidades de información.

Su antecedente más antiguo se remonta a Herman Hollerit, el primer ingeniero estadístico de la historia, quien inventó una computadora llamada máquina automática perforadora de tarjetas en 1884 que facilitaba y agilizaba enormemente algunos procesos de estadística (anteriormente en Estados Unidos, al realizar el censo en 1880, se tuvo que esperar 7 años para obtener los resultados pero al emplear la máquina perforadora en el censo de 1890, ésta arrojó los resultados en 2.5 años, donde además se podían obtener datos importantes como el número de nacimientos, población infantil y número de familias).

En la década de los 50's se da origen a las cintas magnéticas, las cuales ayudaron a cubrir las necesidades de la información industrial, es por este medio que se comienza la automatización; por otra lado, los discos dieron origen a las bases de datos, de red y jerárquicas, ya que para los programadores era más sencillo manipular y guardar las estructuras de los datos como listas y árboles.

En la década de los 60's, se caracteriza por el almacenamiento de disco, las bases de datos jerárquicos y de red.

En la década de los 70's, Edgar Codd definió el modelo relacional y público, una serie de reglas para la evaluación de administradores de sistemas de datos. Es así como surgen las bases de datos relacionales que se caracterizan por ser transaccionales, estables, escalables y de multiplataforma.

En la década de los 80's, cuando las bases de datos relacionales con su sistema de tablas, filas y columnas lograron competir con las bases de datos jerárquicas y de red debido a que el nivel de programación era bajo y mucho más fácil. En esta época comenzaron las investigaciones paralelas y distribuidas como las bases de datos orientadas a objeto.

En la década de los 90's se crea el lenguaje llamado SQL para la toma de decisiones, el cual emplea un lenguaje de consultas estructurado que analiza grandes cantidades de información (Figura 23).

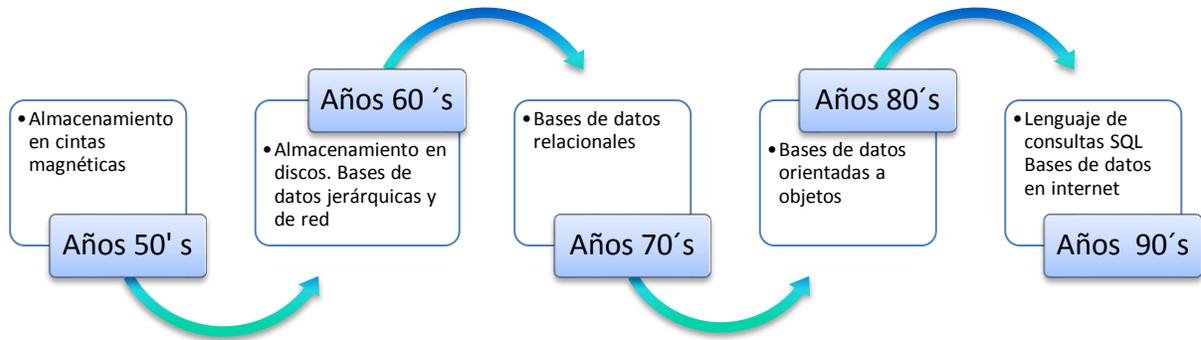


Figura 23. Historia de las bases de datos

2.5.6.2 Principales características de las bases de datos

- Dependencia lógica y física de los datos
- Redundancia mínima
- Acceso concurrente por parte de múltiples usuarios
- Integridad de los datos
- Consultas complejas optimizadas
- Seguridad de acceso y auditoría
- Respaldo y recuperación
- Acceso a través de lenguajes de programación estándar

Entre las ventajas de trabajar con bases de datos, podemos encontrar las siguientes:

Control sobre la redundancia de datos: El sistema almacena varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio. En los sistemas de base de datos los ficheros están integrados por lo cual no almacena varias copias de los mismos datos.

Consistencia de datos: Eliminar o controlar las redundancias de datos reduce en grandes medidas las inconsistencias. Si un dato está almacenado una vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un registro está duplicado y el sistema conoce esta redundancia, el mismo sistema se encarga de garantizar las copias consistentes.

Compartición de datos: La información puede ser compartida por todos los usuarios que estén autorizados.

Mantenimiento de estándares: Establecen formatos de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

Mejora en la integridad de datos: Se refiere a la validez y la consistencia de los datos almacenados. Se maneja mediante restricciones o reglas para que no se puedan violar.

Mejora en la seguridad: Es la protección de las bases de datos frente a los usuarios no autorizados y así los datos podrían encontrarse vulnerables.

Mejora en la accesibilidad a los datos: Proporciona accesibilidad de la información.

Mejora en la productividad: El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel.

Mejora en el mantenimiento: Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados.

Aumento de la concurrencia: Si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o se pierda la integridad. La mayoría de los Sistemas de Gestión de Base de Datos (SGBD) validan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

Mejora en los servicios de copias de seguridad: Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos.

Desventajas de las bases de datos

Complejidad: Conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos.

Vulnerable a los fallos: El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ello que deben tenerse copias de seguridad (Backups).

2.5.6.3 Tipos de base de datos

MySQL: Es un sistema de base de datos relacional con licencia GPL basada en un servidor. Se caracteriza por su rapidez y por permitir el acceso de múltiples usuarios al mismo tiempo.

PostgreSql y Oracle Sun: Sistemas de bases de datos poderosos, recomendables para grandes volúmenes, suele usarse en intranets y sistemas grandes.

Access: Es una base de datos desarrollada por Microsoft. Esta base de datos, debe ser creada bajo el programa Access, el cual crea un archivo .mdb con la estructura ya explicada.

Microsoft SQL Server: Es una base de datos más potente que Access desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de información.

Capítulo 3

Desarrollo del sistema de la
Microfinanciera

3.1 Definición de requerimientos

Con la finalidad de satisfacer las necesidades del cliente se puntualizan los requerimientos en los siguientes puntos:

- Minimizar errores de captura por parte de los usuarios así como evitar omisiones de datos obligatorios
- Erradicar la falta de objetividad por parte de los ejecutivos en cuanto a la asignación de créditos
- Agilizar el proceso de aceptación de un crédito
- En casos extraordinarios, los créditos que no fueron autorizados deberán mostrar el motivo por el que han sido rechazados
- Garantizar el mantenimiento constante de los registros en el sistema
- Garantizar la escalabilidad del sistema
- Garantizar la usabilidad de la interfaz hacia el sistema
- Garantizar la integridad de la información utilizada
- Minimizar costos en la captura de información así como de compartir información con diferentes sucursales
- Minimizar tiempo en los procesos de alta de cliente y crédito
- Garantizar la disponibilidad del sistema en cualquiera de las sucursales de la *Microfinanciera*

Para ello se requiere de la creación de una interfaz que permita la captura de los datos de clientes, cuentas y créditos de la *Microfinanciera* que además almacene la información y la valide.

Conceptualización

La *Microfinanciera* tiene como objetivo dar servicios de alta de clientes, cuentas, créditos y ofrecer los servicios de consultas/modificación respectivas, sin embargo se debe considerar que se requiere automatizar las autorizaciones de los créditos.

Se debe garantizar la disponibilidad de los datos, así como del sistema.

Con la implementación del sistema en la *Microfinanciera*, se pretende optimizar el funcionamiento actual dentro de ella. Por optimización, debe entenderse que se busca un incremento en los beneficios tanto para la empresa como para sus clientes y los procesos que se llevan a cabo dentro de la empresa: alta y manejo de información de clientes, cuentas y créditos relacionados entre sí.

Para alcanzar este objetivo, se desarrollará un sistema que consistirá en la creación de:

- Interfaz de usuario

Son las vistas con la que interactuarán los usuarios para introducir, modificar y consultar información dentro del sistema

Debiendo cumplir con las siguientes características

- Fácil manejo
- Accesibilidad
- Seguridad

- Base de datos

En ella se guardarán los datos que necesita la empresa para realizar sus actividades cotidianas (asignación de préstamos)

Debiendo cumplir con las siguientes características

- Robusta
- Accesibilidad
- Seguridad
- Disponibilidad
- Mantenimiento económico

- Web Service

Será el encargado de realizar el intercambio de información entre la interfaz y la base de datos. Debiendo cumplir con la iteración con diversas interfaces

Análisis de condiciones

El sistema sólo permitirá creaciones de créditos diarios no mayor a 1,000 registros por sucursal.

Se deberá de contar con un servidor capaz de respaldar las aplicaciones necesarias para cada sucursal que será definido de acuerdo a los requerimientos de la *Microfinanciera*.

No está dentro del alcance la administración del crédito para ello se requiere de contar con algún sistema bancario.

Debido a que el crecimiento es variable y depende de la población y sus necesidades, el proyecto se comenzará con software de licencia gratuita y de menor capacidad. Sin embargo, en el caso de requerir mayor capacidad se debe garantizar que la escalabilidad es posible mediante nuevos desarrollos y no de sustitución del actual a realizar.

Análisis de riesgos

- Interfaz de usuario:
 - Cada perfil cuenta con características específicas

- El ejecutivo cuenta con las posibilidades de ofrecer y capturar y/o modificar las cuentas, clientes y créditos
 - El analista de crédito puede autorizar un crédito en caso de ser una situación excepcional
- Procesamiento:
 - El proceso de captura debe quedar respaldado en un repositorio, al cual podrá accederse en cualquier momento así como también podrá modificarse sin poseer redundancia de la información
- Interfaz:
 - Garantizar que la información se encuentra respalda de manera correcta en la base de datos. Se debe validar la congruencia y los datos mismos capturados que van a guardarse en la base

3.1.1 Planificación del proyecto

En el diseño del proyecto es importante dar una estimación de tiempos para conocer el alcance de tiempos y costos. En la planeación del proyecto (Figura 24) se detalla los tiempos necesarios para cada etapa del desarrollo y de esta manera dar a conocer al usuario qué tan viable es el proyecto. Existe una interrelación entre costos y tiempos ya que el cliente busca un menor costo y tiempo para minimizar el impacto a su sistema sin perder la calidad del mismo.

En este proyecto se ha puesto como una meta minimizar costos y tiempos, por ello es importante dar una correcta estimación como en todo proyecto.

Etapa	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Definición de requerimientos	8 días	lun 05/01/15	mié 14/01/15	
2	Planeación del proyecto	4 días	lun 05/01/15	jue 08/01/15	
3	Análisis de requisitos del software	4 días	vie 09/01/15	mié 14/01/15	2
4	Diseño del sistema y del software	30 días	jue 15/01/15	mié 25/02/15	1
5	Diseño de base de datos	8 días	jue 15/01/15	lun 26/01/15	
6	Clientes	2 días	jue 15/01/15	vie 16/01/15	
7	Cuenta	2 días	lun 19/01/15	mar 20/01/15	6
8	Créditos	2 días	mié 21/01/15	jue 22/01/15	7
9	Otras	2 días	vie 23/01/15	lun 26/01/15	8
10	Diseños de Web Service	6 días	mar 27/01/15	mar 03/02/15	
11	Métodos de clientes y cuentas	2 días	mar 27/01/15	mié 28/01/15	9
12	Método de base de datos	2 días	jue 29/01/15	vie 30/01/15	11
13	Método de utilitarios	2 días	lun 02/02/15	mar 03/02/15	12
14	Diseño de interfaz de usuario	16 días	mié 04/02/15	mié 25/02/15	
15	Interfaz de clientes	4 días	mié 04/02/15	lun 09/02/15	13
16	Interfaz de cuentas	4 días	mar 10/02/15	vie 13/02/15	15
17	Interfaz de créditos	4 días	lun 16/02/15	jue 19/02/15	16
18	Interfaz de otros	4 días	vie 20/02/15	mié 25/02/15	17

19	Implementación y pruebas del sistema	82 días	jue 26/02/15	vie 19/06/15	4
20	Desarrollo de Base de datos	12 días	jue 26/02/15	vie 13/03/15	5
21	Clientes	3 días	jue 26/02/15	lun 02/03/15	18
22	Cuenta	3 días	mar 03/03/15	jue 05/03/15	21
23	Créditos	3 días	vie 06/03/15	mar 10/03/15	22
24	Otras	3 días	mié 11/03/15	vie 13/03/15	23
25	Desarrollo de Web Service	30 días	lun 16/03/15	vie 24/04/15	10
26	Métodos de clientes y cuentas	10 días	lun 16/03/15	vie 27/03/15	24
27	Método de base de datos	10 días	lun 30/03/15	vie 10/04/15	26
28	Método de utilitarios	10 días	lun 13/04/15	vie 24/04/15	27
29	Desarrollo de Interfaz de usuario	40 días	lun 27/04/15	vie 19/06/15	14
30	Interfaz de clientes	10 días	lun 27/04/15	vie 08/05/15	28
31	Interfaz de cuentas	10 días	lun 11/05/15	vie 22/05/15	30
32	Interfaz de créditos	10 días	lun 25/05/15	vie 05/06/15	31
33	Interfaz de otros	10 días	lun 08/06/15	vie 19/06/15	32
34	Integración y pruebas unitarias del sistema	38 días	lun 22/06/15	mié 12/08/15	19
35	Integración del Base de datos, Web Service e Interfaz del sistema	8 días	lun 22/06/15	mié 01/07/15	19
36	Pruebas unitarias	30 días	lun 22/06/15	vie 31/07/15	
37	Conexión Web Service	5 días	lun 22/06/15	vie 26/06/15	
38	Conexión de interfaz	5 días	lun 29/06/15	vie 03/07/15	37
39	Pruebas integrales	20 días	lun 06/07/15	vie 31/07/15	38
40	Funcionamiento y mantenimiento	10 días	jue 13/08/15	mié 26/08/15	34

Figura 24 Plan de trabajo

Diagrama de Gantt

Para exponer los tiempos de manera gráfica y que sea entendible para el usuario se hace uso del diagrama de Gantt (Figura 25). Es importante visualizar las dependencias para conocer el alcance del desarrollo.

La planificación del proyecto se basó en la metodología lineal debida a que en la actualidad esta metodología resulta ser la más adecuada por su sencillez y adaptabilidad a cambios.

Como se puede observar en el diagrama de Gantt, se cuenta con tiempos para la definición de requerimientos, diseño del sistema y del software, implementación y prueba de unidades, integración y prueba del sistema además de funcionamiento y mantenimiento.

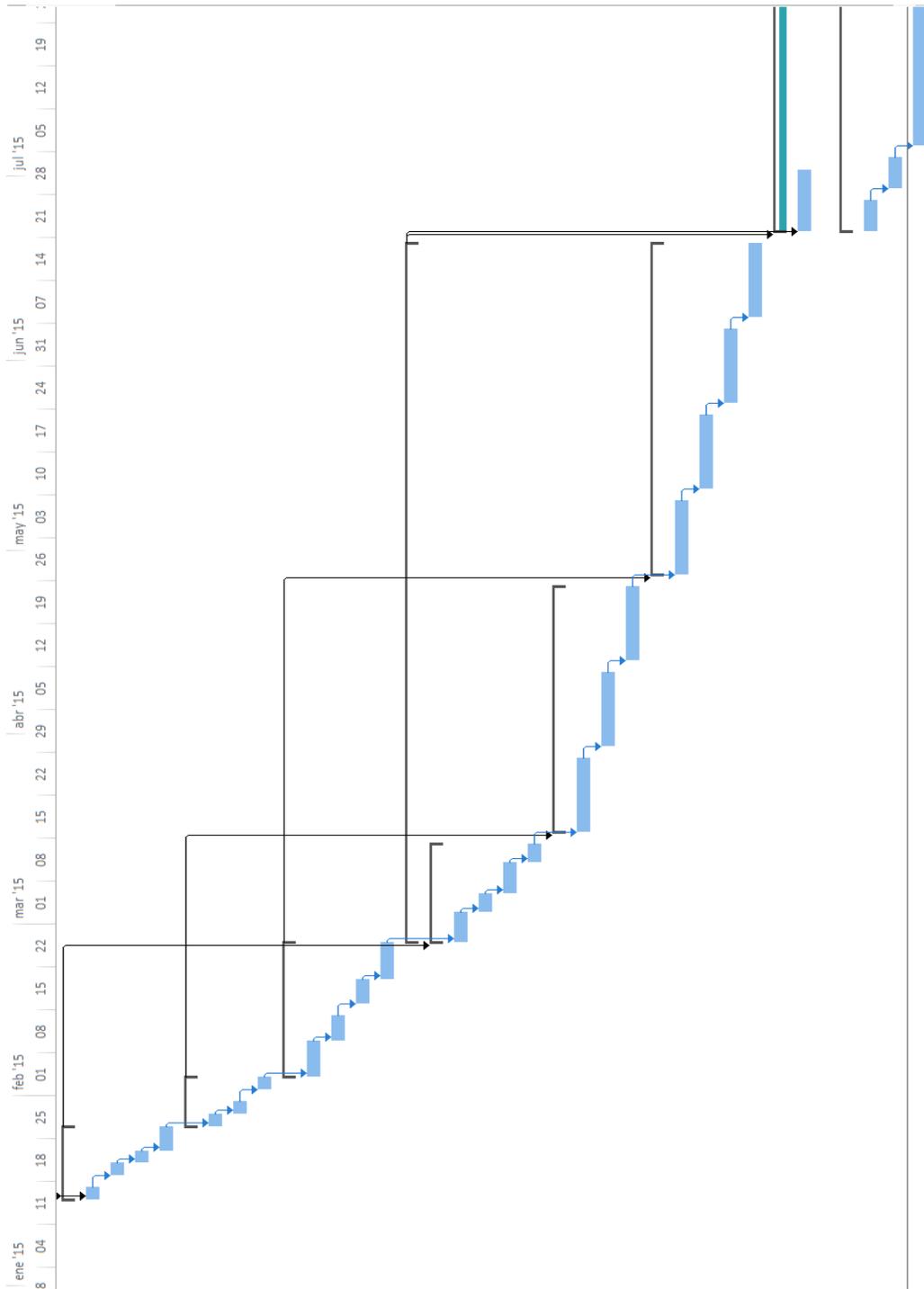


Figura 25 Diagrama de Gantt

3.1.2 Análisis de requisitos del software

Para cumplir con las necesidades de software se requiere:

- Sistema operativo Windows XP, Vista, u otra versión
Para soportar los entornos de desarrollo de interfaces como el MySQL Workbench o Netbeans
- Memoria mayor a 1Gb RAM y 40Gb en disco duro
Garantizar la velocidad y tiempo de respuesta para cada consulta o inserción de dato
- MySQL Workbench
Necesidad de un almacén de datos accesible, de bajo costo y fácil administración
- NetBeans
Cumplir con la interfaz de usuarios y almacén de la información, de una manera robusta y fácil de manejar
- Java jdk 7
Garantizar la comunicación de cada elemento
- Visual Studio 5 u otra versión
Desarrollo de la interfaz de usuario con fácil manejo al usuario y desarrolladores
- Glassfish
Garantizar la comunicación entre los elementos del desarrollo, servidor con toda la seguridad de su comunicación

3.2 Diseño del sistema y del software

3.2.1 Arquitectura del sistema

El diseño arquitectónico se divide en 3 secciones como puede apreciarse en la Figura 26:

- a) La interfaz de usuario se encarga de dar un acceso al usuario al sistema de manera amigable (interfaz Web)
- b) La sección lógica es donde se concentra la lógica del negocio, la cual se encontrará dentro del Web Service y en el diseño de la interfaz Web
- c) La sección base de datos en donde se administrarán los registros ingresados

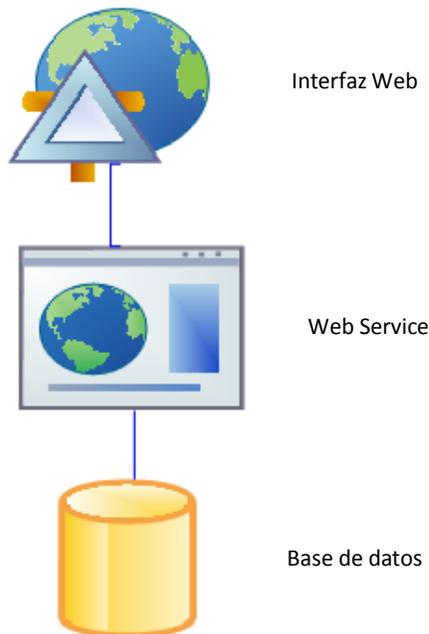


Figura 26. Diseño arquitectónico

3.2.2 Estructura de datos

El usuario requiere de una base de datos que sea capaz de almacenar de manera fácil y con poca capacitación los diferentes datos proporcionados.

Para la creación de la bases de datos capaz de almacenar la información necesaria para la *Microfinanciera* se denominará *FinancieraBD*. Con los siguientes tablas capaces de almacenar la información requerida por la institución.

- *cliente*
- *cuenta*
- *credito*
- *usuario*
- *Sucursal*
- *TipoProducto*
- *Documentacion*

Para la *Microfinanciera* es necesario contar con una base de datos capaz de almacenar los clientes existentes y nuevos dentro del sistema para tener una buena administración de los datos ofrecidos por cada cliente y para la institución es importante prepararse para un captar un mayor número de clientes en un futuro. Para ello se creó una tabla llamada *cliente*.

Campo	Requerido	Longitud	Tipo de dato	Descripción
IdCliente	Mandatorio	11	Numérico	Identificador único del cliente
Nombre	Mandatorio	45	Alfanumérico	Nombre del cliente
ApellidoPa	Mandatorio	45	Alfanumérico	Apellido paterno del cliente
ApellidoMa	Mandatorio	45	Alfanumérico	Apellido materno del cliente
Edad	Mandatorio	3	Numérico	Edad del cliente
Sexo	Mandatorio	10	Alfanumérico	Sexo del cliente
Nacionalidad	Mandatorio	45	Alfanumérico	Nacionalidad del cliente
IdDocumentación	Mandatorio	45	Alfanumérico	Documentación entregada
NulIdentificación	Mandatorio	45	Alfanumérico	Número de documentación
Calle	Mandatorio	45	Alfanumérico	Calle declarada por el cliente
NoExt	Mandatorio	4	Numérico	Número exterior del cliente
NoInt	Mandatorio	4	Numérico	Número interior del cliente
CoPostal	Mandatorio	8	Numérico	Código postal del cliente
Delegación	Mandatorio	45	Alfanumérico	Nombre de la delegación del cliente
Municipio	Mandatorio	45	Alfanumérico	Nombre del municipio del cliente
IdSucursal	Mandatorio	45	Numérico	Número de la sucursal
Estado	Mandatorio	45	Alfanumérico	Estado en donde reside el cliente
País	Mandatorio	45	Alfanumérico	País en donde reside el cliente
FechaIngreso	Mandatorio	8	Numérico	Fecha de apertura
CURP	Mandatorio	24	Alfanumérico	Clave única de registro de población
FechaNa	Mandatorio	8	Numérico	Fecha de nacimiento del cliente
EstadoCivil	Opcional	10	Alfanumérico	Estado civil del cliente
RFC	Opcional	18	Alfanumérico	Número de referencia fiscal
NoCel	Opcional	15	Numérico	Número de celular del cliente
Email	Opcional	45	Alfanumérico	Correo electrónico del cliente
TipIdent	Opcional	20	Alfanumérico	Tipo de identificación
Nolenti	Opcional	20	Alfanumérico	Número de identificación

En la base de datos se crea una tabla llamada *cuenta*, donde se almacenarán las cuentas asignadas a los créditos.

Campo	Requerido	Longitud	Tipo de dato	Descripción
IdCliente	Mandatorio	11	Numérico	Identificador del cliente
IdCuenta	Mandatorio	11	Numérico	Identificador de la cuenta
OfficerAccount	Mandatorio	15	Alfanumérico	Ejecutivo de cuenta que realiza la apertura
TypeProduct	Mandatorio	20	Alfanumérico	Tipo de producto que se ofrece al cliente
Currency	Mandatorio	4	Alfanumérico	Moneda con el cual se está trabajando
FechaApe	Mandatorio	8	Numérico	Fecha de apertura de la cuenta
NomBene	Opcional	45	Alfanumérico	Nombre de beneficiario
ApPaBene	Opcional	45	Alfanumérico	Apellido paterno beneficiario
ApMaBene	Opcional	45	Alfanumérico	Apellido materno beneficiario
ChargeOpen	Opcional	5	Alfanumérico	Comisión de apertura
PorcRelac	Opcional	5	Alfanumérico	Porcentaje de relación
DepEffect	Opcional	3	Alfanumérico	Deposito en efectivo
ReEffect	Opcional	3	Alfanumérico	Retiro en efectivo
TransEffect	Opcional	3	Alfanumérico	Transferencias recibos
DepOtros	Opcional	3	Alfanumérico	Depósitos otros
PagoCheques	Opcional	3	Alfanumérico	Pago de cheques

Para almacenar los créditos creados en la interfaz se genera la tabla *credito*

Campo	Requerido	Longitud	Tipo de dato	Descripción
IdCrédito	Mandatorio	11	Numérico	Identificador del crédito
IdCuenta	Mandatorio	11	Numérico	Identificador de la cuenta
IdCliente	Mandatorio	11	Numérico	Identificador del cliente
TypeProduct	Mandatorio	20	Alfanumérico	Tipo de producto que se ofrece al cliente
Currency	Mandatorio	4	Alfanumérico	Moneda con la cual se está trabajando
Monto	Mandatorio	14	Numérico	Monto del crédito
Plazo	Mandatorio	14	Numérico	Plazo del crédito
TasaInt	Mandatorio	4	Alfanumérico	Tasa de interés del préstamo
TipoTasa	Mandatorio	10	Alfanumérico	Tipo de tasa del préstamo
FreqPago	Mandatorio	10	Alfanumérico	Frecuencia de pago del préstamo
LineaPermitida	Mandatorio	2	Alfanumérico	Línea permitida para el cliente
EnviDom	Opcional	2	Alfanumérico	Envío de documentos al cliente

Para la definición de los productos se crea la tabla *TipoProducto*.

Campo	Requerido	Longitud	Tipo de dato	Descripción
IdTipoProducto	Mandatorio	18	Numérico	Identificador de tipo de producto
Descripción	Mandatorio	45	Numérico	Descripción tipo del producto

En la definición de perfiles y permisos se genera la tabla *usuario*.

Campo	Requerido	Longitud	Tipo de dato	Descripción
IdUsuarios	Mandatorio	11	Numérico	Identificador del usuario
NombreEje	Mandatorio	45	Alfanumérico	Nombre del ejecutivo
ApellidoPaEje	Mandatorio	45	Alfanumérico	Apellido paterno del ejecutivo
ApellidoMaEje	Mandatorio	45	Alfanumérico	Apellido materno del ejecutivo
IdSucursal	Mandatorio	45	Alfanumérico	Numero de sucursal

Para el control de los documentos que se pueden manejar se crea una tabla de *Documentación*

Campo	Requerido	Longitud	Tipo de dato	Descripción
IdDocumentación	Mandatorio	11	Numérico	Identificador de la documentación a entregar
Descripcion	Mandatorio	45	Alfanumérico	Descripción documentación a entregar
Nuldetificacion	Opcional	45	Alfanumérico	Se podría almacenar el número de identificación
IdCredito_	Mandatorio	11	Numérico	Id del crédito relacionado

El control de registros por sucursal para ello se usa una tabla llamada *Sucursal*.

Campo	Requerido	Longitud	Tipo de dato	Descripción
IdSucursal	Mandatorio	11	Numérico	Identificador de la sucursal
Ubicación	Mandatorio	11	Alfanumérico	Descripción de la localización de sucursal
Descripción	Mandatorio	45	Alfanumérico	Descripción de la sucursal donde se realiza la captura
IdUsuario	Mandatorio	11	Numérico	Identificador del usuario con facultades
Ubicación	Mandatorio	45	Alfanumérico	Ubicación geográfica de la sucursal

La Figura 27 presenta el modelo entidad-relación de la base de datos mostrando todas relaciones.

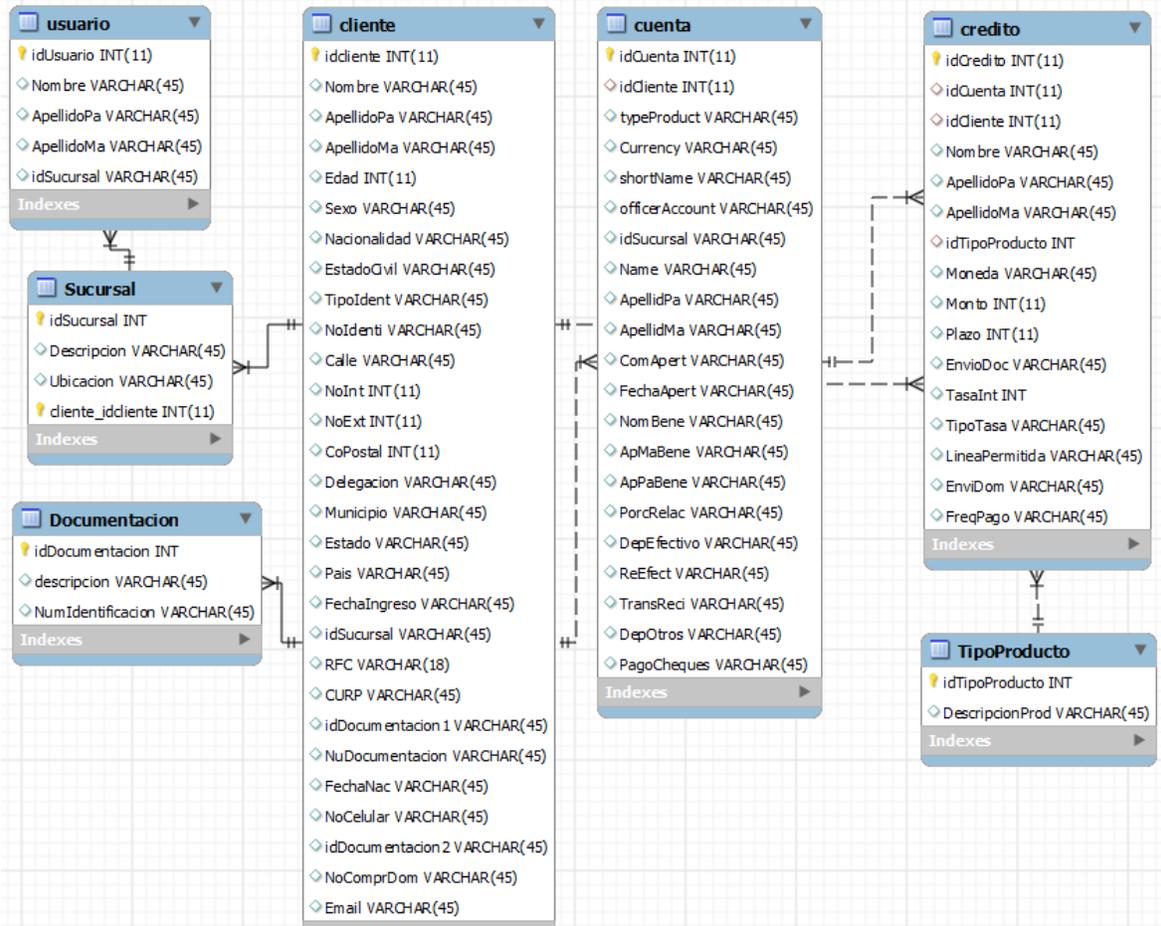


Figura 27. Diagrama Entidad-Relación.

3.2.3 Detalle procedimental

3.2.3.1 Diseño de Web Service

El Web Service se caracteriza por facilitar la comunicación con diferentes interfaces para la *Microfinanciera*; esto debe garantizar que, cuando se requiera un crecimiento, no haya necesidad de migrar a otra tecnología; es decir, que sea escalable.

Mediante diversos formularios se capturan los datos necesarios para generar nuevos clientes, cuentas y créditos (Figura 28).

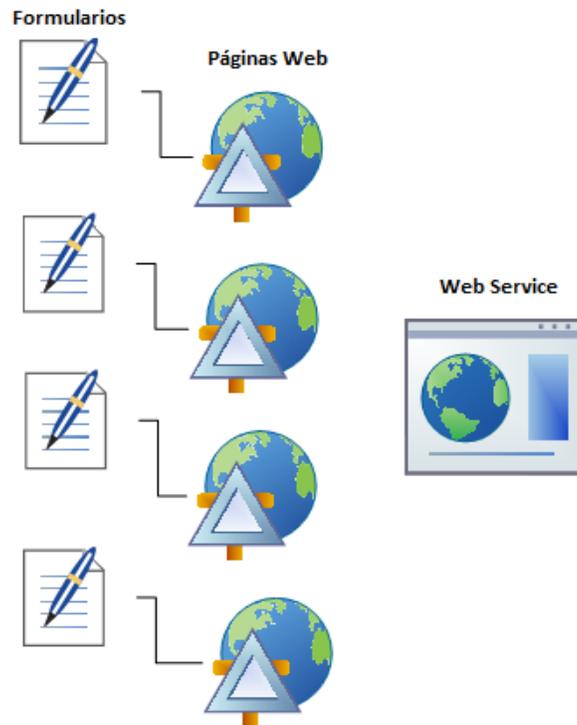


Figura 28 Web Service para *Microfinanciera*

La información capturada en los formularios es guardada en una base de datos, en este caso MySQL, para poder contar con una administración adecuada de la misma. De esta manera se garantiza la disponibilidad de los datos en todo momento.

Mediante el Web Service se podrá realizar la conexión a la base de datos (Figura 29) haciendo uso de la función `DriverManager.getConnection(url, login, password)` y el drive `Class.forName("com.mysql.jdbc.Driver")` que se encuentra almacenado en el jar MySQL JDBC Driver `mysql-connector-java-5.1.23-bin.jar` cargado en el IDE de NETBEANS o bien se podrá configurar de forma manual en el entorno de trabajo de la siguiente forma:

```

PATH=${PATH}:/usr/local/mysql/bin
CLASSPATH=$CLASSPATH:/System/Library/Java/Extensions/
CLASSPATH=$CLASSPATH:/System/Library/Frameworks/JavaVM.framework/Versions/1.5.
0/Classes:/Library/Java/Extensions/
export CLASSPATH

```

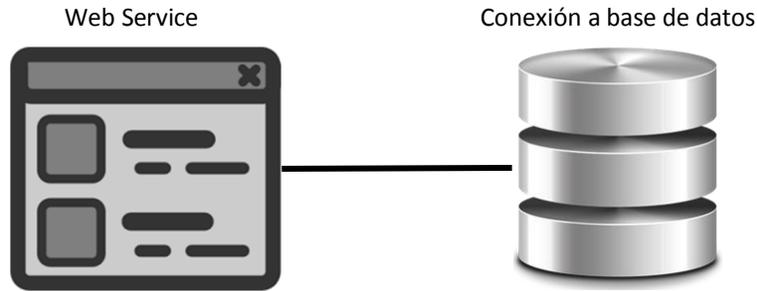


Figura 29 Conexión a base de datos

Para cada proceso bancario se deberá crear un método de acuerdo al objetivo que se desee realizar para la comunicación (Figura 30).

- Alta de clientes
@WebMethod(operationName = "AltaCliente")
- Alta de cuenta
@WebMethod(operationName = "AltaCuenta")
- Alta de crédito
@WebMethod(operationName = "AltaCredito")
- Consulta de clientes
@WebMethod(operationName = "ConsultaCliente")

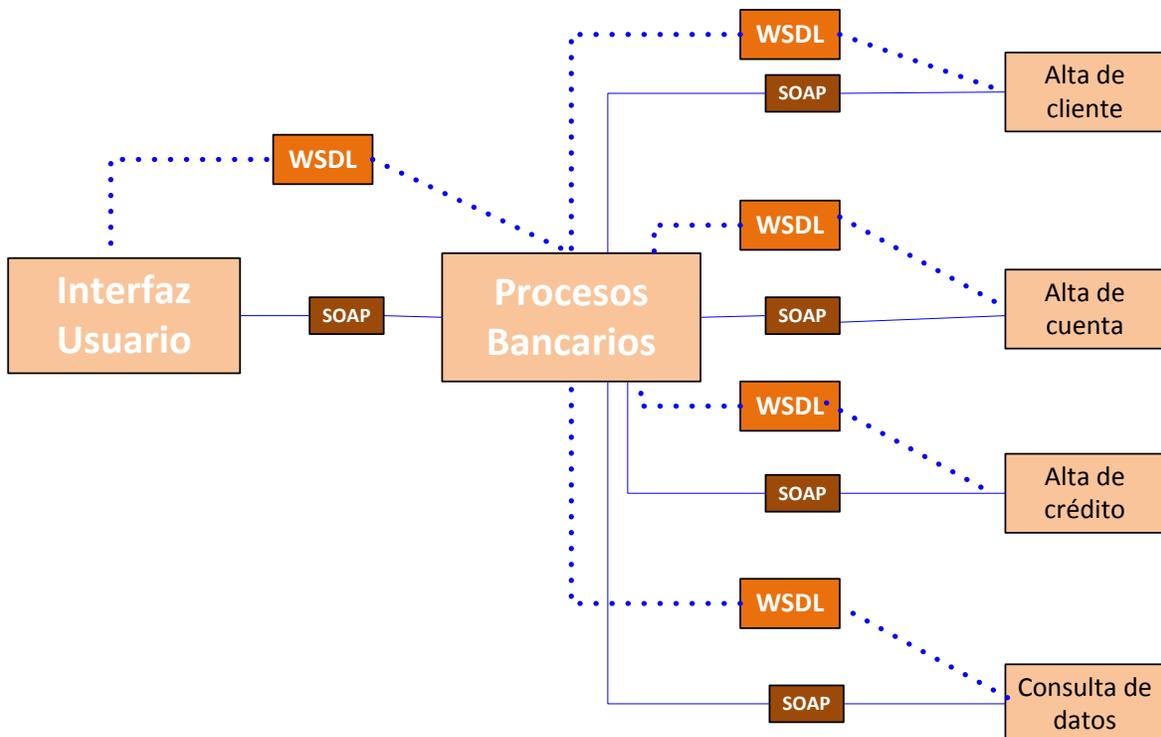


Figura 30 Sistema de la microfinanciera

Para garantizar la correcta captura de los datos de las cuentas y clientes se consideran algunas validaciones (Figura 31).

Casos crédito correspondiente:

- Monto de préstamo es menor al valor de la garantía
- El cliente ya cuenta con préstamos anteriores y no incurrió en falta
- La línea de crédito es mayor \$20,000.00
- Monto del préstamo es menor de \$5,000.00
- No cuenta con observaciones

Casos en los cuales se requiere de autorización.

- El monto del préstamo es mayor a \$1,000,000.00
- El monto del préstamo es mayor a la garantía o no cuenta con garantía
- El cliente ya cuenta con préstamos
- El cliente ha tenido préstamos vencidos
- El cliente cuenta con observaciones

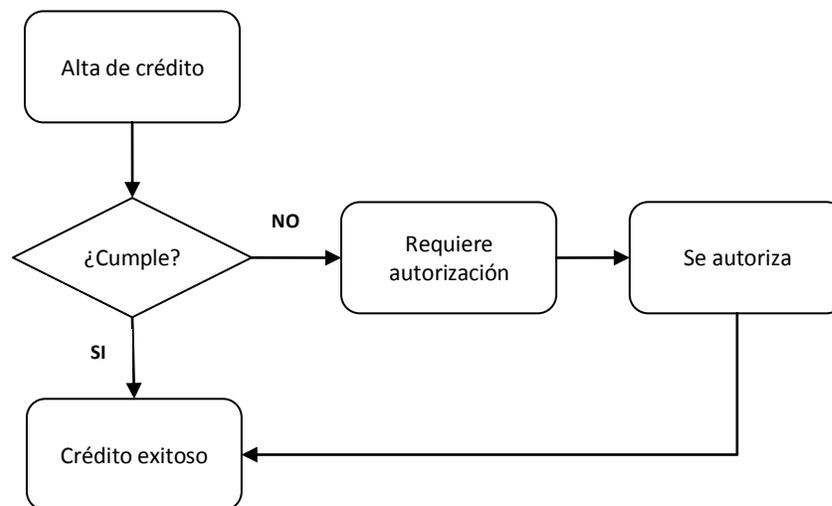


Figura 31 Flujo de aceptación de crédito

3.2.4 Interfaces

Mediante el uso de la IDE Visual Studio se crean las interfaces finales del usuario.

Interfaces de usuarios (Figura 32), los campos reflejados serán los campos mencionados en el diseño de la base de datos:

- Alta de cuenta

- Alta de cliente
- Alta de crédito

Mediante una consulta se pueden modificar los registros:

- Consulta de cuenta
- Consulta de cliente
- Consulta de crédito
- Eliminar registros



Figura 32 Interfaz de usuario

Validaciones que se desarrollan en la interfaz:

- Longitud de los campos
- Campos obligatorios y opcionales

3.3 Implementación y prueba de unidades

La implementación es el paso en donde se desarrolla cada uno de los diseños anteriormente planteados. Se comienza con la base de datos debido a que es la plataforma de todo el desarrollo.

Haciendo uso de diferentes lenguajes e IDE's, se realiza la comunicación entre la base de datos, el Web Service y la interfaz de usuario C#.

3.3.1 Implementación de la base de datos

Actualmente MySQL permite crear de diversas formas las bases de datos que se utilizan en un sistema, pueden realizarse a través de código o mediante diagramas que son más amigables para el desarrollador.

Para crear la tabla de *cuenta*:

```
CREATE TABLE IF NOT EXISTS `finacierabd`.`cuenta` (  
  `idCuenta` INT(11) NOT NULL ,  
  `idCliente` INT(11) NULL DEFAULT NULL ,  
  `typeProduct` VARCHAR(45) NULL DEFAULT NULL ,  
  `Currency` VARCHAR(45) NULL DEFAULT NULL ,  
  `shortName` VARCHAR(45) NULL DEFAULT NULL ,  
  `officerAccount` VARCHAR(45) NULL DEFAULT NULL ,  
  `idSucursal` VARCHAR(45) NULL DEFAULT NULL ,  
  `Name` VARCHAR(45) NULL DEFAULT NULL ,  
  `ApellidoPa` VARCHAR(45) NULL DEFAULT NULL ,  
  `ApellidoMa` VARCHAR(45) NULL DEFAULT NULL ,  
  `ComApert` VARCHAR(45) NULL DEFAULT NULL ,  
  `FechaApert` VARCHAR(45) NULL DEFAULT NULL ,  
  `NomBene` VARCHAR(45) NULL DEFAULT NULL ,  
  `ApMaBene` VARCHAR(45) NULL DEFAULT NULL ,  
  `ApPaBene` VARCHAR(45) NULL DEFAULT NULL ,  
  `PorcRelac` VARCHAR(45) NULL DEFAULT NULL ,  
  `DepEfectivo` VARCHAR(45) NULL DEFAULT NULL ,  
  `ReEfect` VARCHAR(45) NULL DEFAULT NULL ,  
  `TransReci` VARCHAR(45) NULL DEFAULT NULL ,  
  `DepOtros` VARCHAR(45) NULL DEFAULT NULL ,  
  `PagoCheques` VARCHAR(45) NULL DEFAULT NULL ,  
  PRIMARY KEY (`idCuenta`),  
  INDEX `idCuenta` (`idCuenta` ASC),  
  CONSTRAINT `idCliente`  
    FOREIGN KEY (`idCuenta` )  
    REFERENCES `finacierabd`.`cliente` (`idCuenta` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1  
COLLATE = latin1_swedish_ci
```

Para crear tablas mediante el uso de diagramas (Figura 33):

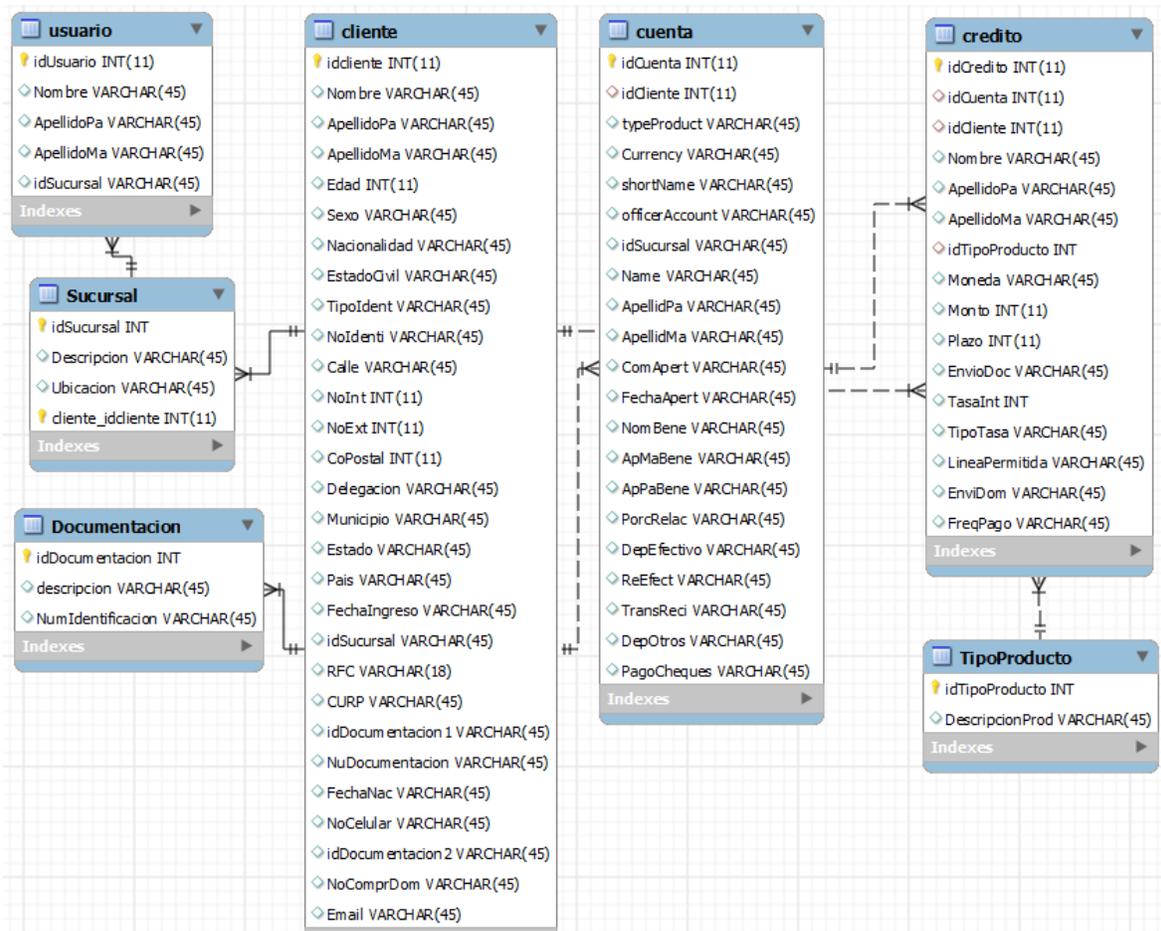


Figura 33 Tablas por diagramas

Es importante visualizar que tenemos llaves primarias y llaves foráneas, haciendo la relación de datos.

3.3.2 Implementación de Web Service

Con la finalidad de tener un sistema escalable y eficiente, se crea un Web Service que sirve para realizar la comunicación entre la base de datos y la interfaz de usuario.

Para realizar la comunicación desde el Web Service, se debe realizar la configuración del servidor de base de datos (SQL Service) previamente.

El código de la comunicación queda como sigue:

```

Conectar()
    Class.forName("com.mysql.jdbc.Driver")
    DriverManager.getConnection(url, login, password);

```

```

        if (conn != null){
            st=conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
        }
        Si no es correcto se captura el error y regresa
    }

```

Código de conexión de bases de datos

En donde los datos a configurar son el puerto, la URL, usuario y la contraseña de la base de datos creada previamente.

Las librerías utilizadas para el desarrollo:

```

package Productos;

import static java.lang.Math.abs;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

```

Método Alta de cliente: Utilizado para la captura de los clientes.

```

@WebMethod(operationName = "AltaCliente")

Declaración de los datos a ingresar

Realiza la conexión a la base de datos: Conexión conector= new conexión();

Realiza la generación del ID único del cliente con la función Random()

Proceso de datos, validación y análisis del resultado de la base de datos

```

Código de captura de clientes

Asimismo dentro del desarrollo del Web Service colocamos la lógica de aceptación de los créditos.

El Web Service se visualiza de la siguiente manera dentro del IDE NETBEANS (Figura 34):

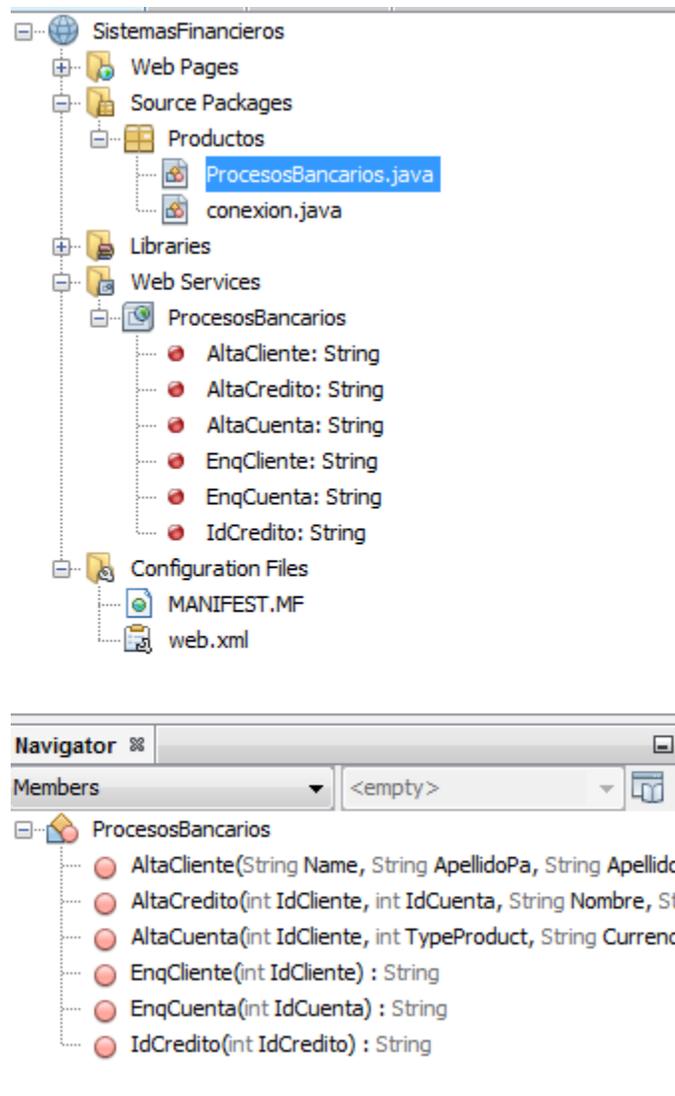


Figura 34 Explorador del proyecto

Las operaciones se realizan a través de la clase llamada ProcesosBancarios.java donde se encuentran los métodos que realizan la inserción de los registros así como las consultas.

3.3.3 Implementación de interfaz de usuarios

Utilizando Visual Studio .Net (C#), se realiza la interfaz para el usuario, la cual es intuitiva y de fácil uso para acelerar el proceso de aprendizaje de los usuario de la *Microfinanciera*.

Se deben crear los métodos para alta de clientes (Figura 35), alta de cuenta y consulta de cuenta, clientes y créditos.

Asimismo contamos con los métodos necesarios para los detalles que serán mostrados al ejecutivo para conocer las características de la *Microfinanciera*. En el caso del proyecto tenemos las páginas *About* y *Default*.

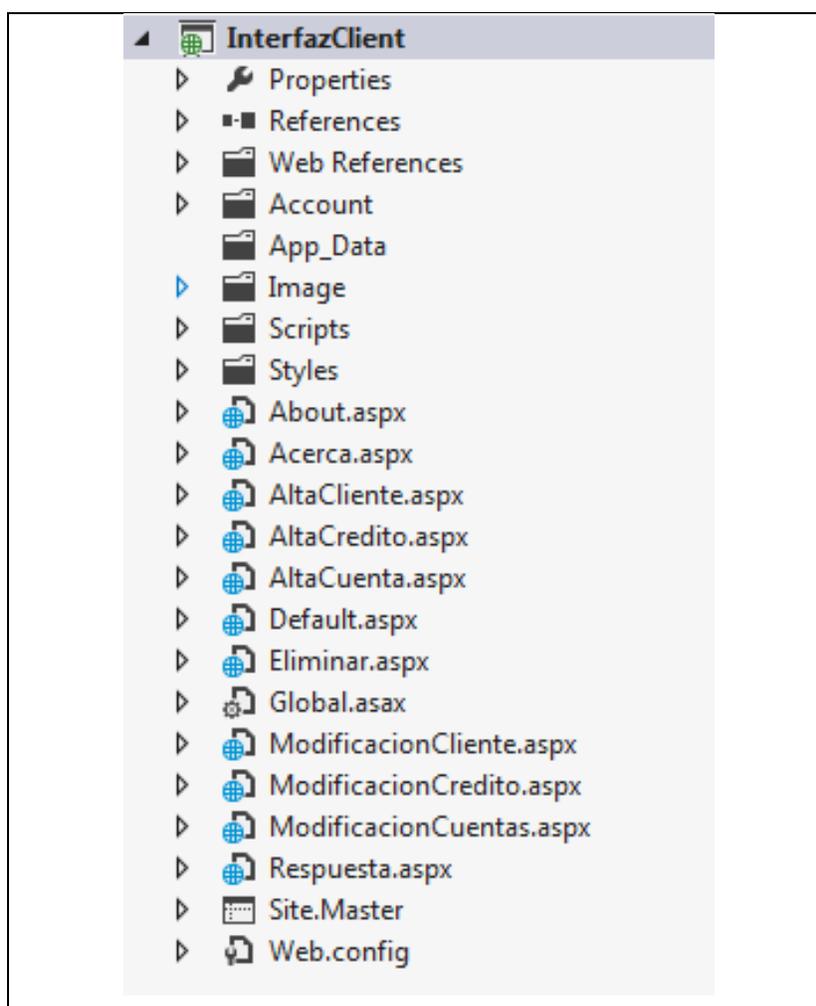


Figura 35 Métodos de servicios

Para facilitar el manejo de las opciones disponibles para el usuario, se crea un menú que ayudará a hacer el proceso de captura de datos más eficiente (Figura 36).

Las opciones disponibles son:

- Alta de cuenta
- Alta de cliente
- Alta de crédito
- Consulta de cuenta
- Consulta de cliente
- Consulta de crédito
- Eliminar registros



Figura 36 Menú principal de la interfaz

La *Microfinanciera* coloca una pantalla en donde brevemente se explica su objetivo y productos que ofrece (Figura 37) para dar a conocer al ejecutivo los detalles de la *Microfinanciera*.



Figura 37 Opción de Quiénes Somos

En ella se realizan las validaciones necesarias para asegurar que es capturada la información mínima indispensable para poder generar los registros que estarán asociados a cada cliente (número de cliente, número de cuenta y número de crédito correspondientes).

```
<div style="float: left; width: 848px; height: 41px;">
    <asp:TextBox ID="FechaIngBox" runat="server"></asp:TextBox>
    <br />
    <asp:RequiredFieldValidator runat="server" ID="RequiredFieldValidator8"
    ErrorMessage="*Fecha Ingreso requerido"
    ControlToValidate="FechaIngBox" ForeColor="Red" />
    <br />
    <br />
</div>
```

Validación de datos obligatorios

Para realizar el alta de un cliente nuevo, es necesario que se capturen algunos datos de manera obligatoria pues de ello depende que se le dé un seguimiento adecuado a la solicitud (Figura 38).

MICROFINANCIERA [Iniciar sesión]

Quiénes somos Menú Principal Acerca de

ALTA DE CLIENTE

Datos del banco

No. de sucursal

Fecha de ingreso

*Fecha ingreso requerido

Datos generales del cliente

Nombre <input type="text"/>	Apellido paterno <input type="text"/>	Apellido materno <input type="text"/>
*Nombre cliente requerido	*Apellido paterno requerido	*Apellido materno requerido
Edad <input type="text"/>	Sexo <input type="text"/>	Estado civil <input type="text"/>
*Edad requerido		
Nacionalidad <input type="text"/>	RFC <input type="text"/>	CURP <input type="text"/>
	*RFC requerido	*CURP requerido
Tipo de identificación <input type="text"/>	Número de identificación <input type="text"/>	Fecha de nacimiento <input type="text"/>
*Tipo identificación requerido	*Número identificación requerido	*Fecha de Nac. requerido

Dirección del cliente

Calle <input type="text"/>	No. interior <input type="text"/>	No. exterior <input type="text"/>
Código postal <input type="text"/>	Delegación <input type="text"/>	Municipio <input type="text"/>
Estado <input type="text"/>	País <input type="text"/>	No. celular <input type="text"/>
Comprobante de domicilio <input type="text"/>	*País requerido	Correo electrónico <input type="text"/>
*Comprobante requerido		
No. comprobante de domicilio <input type="text"/>		
*No. comprobante requerido		

Figura 38 Validación de campos para el alta de cliente

Una vez que se tengan los datos correctos del cliente en el sistema, es necesario asociarle una cuenta en la que se realizarán los movimientos del crédito (desde la disposición hasta el pago de las cuotas). Para ello, también es necesario que sean capturados algunos datos de manera obligatoria a fin de evitar ambigüedades en la asociación de los registros.

Asimismo, cuando va a darse de alta el crédito como tal, deben cumplirse ciertos requisitos para asegurar que el registro no tiene posibilidad de ser asociado con otro cliente por error.

3.4 Integración y prueba del sistema

Una vez que se tiene el desarrollo completo del sistema se verifica que éste funcione de acuerdo a lo previsto.

Para las vistas de creación de cliente, cuenta o crédito, se valida que el sistema impida el alta del registro si no se llenan ciertos campos obligatoriamente. Así, tenemos por ejemplo:

- Cliente.- como mínimo debe incluir nombre completo del cliente (apellido paterno, apellido materno y nombres), género y fecha de nacimiento.
- Cuenta.- obligatoriamente debe ligarse a un número de cliente dado de alta previamente en el sistema.
- Crédito.- para dar de alta un crédito, es necesario que el cliente tenga ya una cuenta asociada.

Para el proceso de captura de cliente se ha diseñado la siguiente interfaz (Figura 39) que cuenta con los campos necesarios y planteando por el requerimiento para el registro de un cliente en la base de la *Microfinanciera*.

MICROFINANCIERA [Iniciar sesión]

Quiénes somos Menú Principal Acerca de

ALTA DE CLIENTE

Datos del banco

No. de sucursal

Fecha de ingreso

Datos generales del cliente

Nombre

Apellido paterno

Apellido materno

Edad

Sexo

Estado civil

Nacionalidad

RFC

CURP

Tipo de identificación

Número de identificación

Fecha de nacimiento

Dirección del cliente

Calle

No. interior

No. exterior

Código postal

Delegación

Municipio

Estado

País

No. celular

Comprobante de domicilio

Correo electrónico

No. comprobante de domicilio

Guardar

Figura 39 Captura de cliente

Para el proceso de captura de cuenta (Figura 40) se muestra al ejecutivo los campos necesarios para el registro de la cuenta, siempre y cuando previamente se haya llenado el número de cuenta.

Figura 40 Pantalla de alta de cuenta

Pantalla de alta de crédito (Figura 41) será capaz de realizar la captura de un crédito en el sistema de la *Microfinanciera* con los campos necesarios.

Figura 41 Alta de crédito

Opción de consulta de datos de cliente (Figura 42):

MICROFINANCIERA [Iniciar sesión]

Quiénes somos | Menú Principal | Acerca de

CONSULTA DE CLIENTE

Datos del banco

No. de sucursal

No. de cliente

Fecha de ingreso

Datos generales del cliente

Nombre

Apellido paterno

Apellido materno

Edad

Sexo

Estado civil

Nacionalidad

RFC

CURP

Tipo de identificación

Número de identificación

Fecha de nacimiento

Dirección del Cliente

Calle

No. interior

No. exterior

Código postal

Delegación

Municipio

Estado

Pais

No. celular

Comprobante de domicilio

Correo electrónico

No. Comprobante de domicilio

Figura 42 Consulta de cliente

Opción de consulta de datos de cuenta (Figura 43) ofrece al ejecutivo la opción de visualizar los datos de una cuenta ya existente y de esta manera poder conocer los detalles de cada cuenta.

MICROFINANCIERA [Iniciar sesión]

Quiénes somos | Menú Principal | Acerca de

CONSULTA DE CUENTA

Número de cuenta

Número de sucursal

Número de cliente

Tipo de cuenta

Nombre

Apellido paterno

Apellido materno

Comisión de apertura

Fecha apertura

Moneda

Ejecutivo de cuenta

Beneficiario

Nombre

Apellido paterno

Apellido materno

Parentezco

% de relación

Perfil de cuenta

Depósito en efectivo

Depósito otros

Retiro en efectivo

Pago de cheques

Transferencias emitidas/recibidas

Figura 43 Consulta de cuenta

Opción de consulta de datos del crédito (Figura 44):



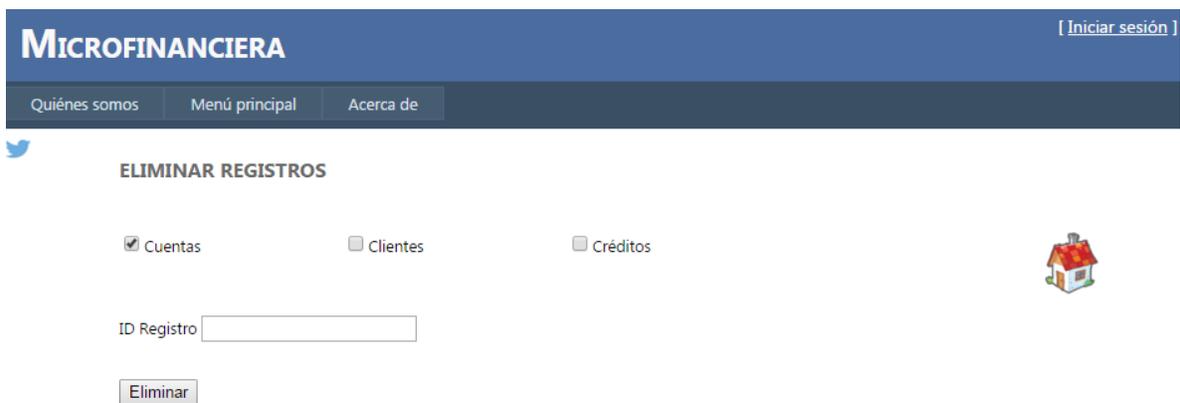
The screenshot shows the 'CONSULTA DE CRÉDITO' (Credit Query) form. At the top, there is a blue header with the text 'MICROFINANCIERA' and a link '[Iniciar sesión]'. Below the header is a navigation bar with three items: 'Quiénes somos', 'Menú principal', and 'Acerca de'. The main content area has a blue Twitter icon on the left and a house icon on the right. The form itself consists of several input fields arranged in a grid:

- Número de crédito
- Número de cuenta
- Número de cliente
- Nombre
- Apellido paterno
- Apellido materno
- Tipo de producto
- Moneda
- Tipo de tasa (with 'FIJA' selected)
- Plazo
- Monto
- Tasa de interés
- Línea permitida
- Frecuencia de pago

At the bottom center of the form is a button labeled 'Actualizar'.

Figura 44 Consulta de crédito

Opción de eliminación de registros (Figura 45):



The screenshot shows the 'ELIMINAR REGISTROS' (Delete Records) form. At the top, there is a blue header with the text 'MICROFINANCIERA' and a link '[Iniciar sesión]'. Below the header is a navigation bar with three items: 'Quiénes somos', 'Menú principal', and 'Acerca de'. The main content area has a blue Twitter icon on the left and a house icon on the right. The form includes three radio buttons for selection:

- Cuentas
- Clientes
- Créditos

Below the radio buttons is an input field labeled 'ID Registro'. At the bottom left of the form is a button labeled 'Eliminar'.

Figura 45 Eliminación de registros

Pantallas con información de la financiera, importantes para los usuarios y clientes (Figura 46):



CRÉDITO:

El crédito es un acto a través del cual una persona (acreedor) confía dinero a otra persona (llamada deudor) por un periodo determinado. Una vez transcurrido el plazo, la persona que recibió el dinero se lo devuelve al acreedor. Usualmente los créditos no son gratuitos, por lo que el deudor, al momento de devolverle el dinero al acreedor o antes, debe agregar un pago adicional o premio al cual se le denomina "interés" y se expresa o se da a conocer a través de la tasa de interés. Una de las principales actividades de los bancos es proporcionar créditos a sus clientes, principalmente personas y empresas. Los créditos también se conocen como "préstamos" o "financiamientos" en el lenguaje bancario.

Las personas (deudores) pueden utilizar el dinero obtenido a través de los créditos según sus necesidades, ya sea para la adquisición de bienes de consumo o para invertirlo. También existen los créditos que se otorgan a las empresas, para que éstas inviertan en bienes intermedios o de capital a fin de mejorar sus procesos de producción y aumentar su producción.

Los bancos ofrecen distintos tipos de crédito para ajustarse a las necesidades de los clientes. Estos créditos se orientan al consumo y a las actividades productivas. Los bancos ofrecen crédito a los hogares a través de tarjetas de crédito, créditos hipotecarios, créditos para la adquisición de bienes de consumo duradero y créditos automotrices. En tanto, las empresas solicitan créditos pyme, microcréditos, de habilitación y avío, refaccionarios, quirografarios, prendarios, e interbancarios y refaccionarios, entre otros.

El deudor puede pagar el crédito a plazos previamente establecidos o bien puede hacer un solo pago al final del periodo por el total del crédito, adicionando el interés que se haya pactado entre el deudor y el acreedor. Por lo general, una vez que se utiliza el crédito no se puede volver a disponer del dinero aunque el deudor ya lo haya pagado.

Referencia Banco de México: <http://www.banxico.org.mx/divulgacion/sistema-financiero/sistema-financiero.html#Serviciosdecredito>

Figura 46 Pantalla "Acerca de"

3.4.1 Algunas pruebas del sistema

Con la finalidad de realizar la verificación de los requerimientos se muestran a continuación algunas pruebas unitarias e integrales.

Interfaz de usuario final

Pruebas unitarias:

Caso de prueba 1: Consumo correcto del Web Service por la interfaz de usuario (en este caso la interfaz de *alta de crédito* como puede observarse en la Figura 47).

Prueba unitaria 1: Conexión exitosa



ALTA DE CRÉDITO

Número de cliente 1000010003	Número de cuenta 1000010003	Envío documento (Si/No) SI
Nombre MARIA DEL CARMEN	Apellido paterno LOPEZ	Apellido materno LOPEZ
Tipo de producto NUEVO	Moneda MXN	Tipo de tasa FIJA
Plazo (Meses) 24	Monto 45000	Tasa de interés 12.3
Línea permitida(Si/No) SI	Frecuencia de pago MENSUAL	

Guardar

Figura 47 Interfaz de alta de crédito

Mediante el log servidor se verifica que la interfaz está consumiendo de manera correcta al Web Service.

```
Información: Webservice Endpoint deployed ProcesosBancarios  
listening at address at http://Catarsys-PC:8080/SistemasFinancieros/ProcesosBancarios.  
Información: Loading application [SistemasFinancieros] at [/SistemasFinancieros]  
Información: SistemasFinancieros was successfully deployed in 2,134 milliseconds.  
Información: Conexión exitosa : jdbc:mysql://localhost:3306/finacierabd  
Información: datos: insert into credito values('-1701618752','1000010003','1000010003',MARIA DEL  
CARMEN','LOPEZ','LOPEZ','NUEVO','MXN','45000','24','SI','12.3','FIJA','SI')
```

Prueba Unitaria 2: Caso no exitoso, el servidor no se encuentra disponible

Mediante el log servidor se verifica que la interfaz no está consumiendo de manera correcta al Web Service.

```
Información: Webservice Endpoint deployed ProcesosBancarios  
listening at address at http://Catarsys-PC:8080/SistemasFinancieros/ProcesosBancarios.  
Información: Loading application [SistemasFinancieros] at [/SistemasFinancieros]  
Información: SistemasFinancieros was successfully deployed in 2,134 milliseconds.  
Información: Hubo un problema al conectar con la base jdbc:mysql://localhost:3306/finacierabd  
Información: datos: insert into credito values('-1701618752','1000010003','1000010003',MARIA DEL  
CARMEN','LOPEZ','LOPEZ','NUEVO','MXN','45000','24','SI','12.3','FIJA','SI')
```

Prueba Unitaria 3: Prueba de creación del Web Service

Mediante la herramienta de prueba del Web Service que nos proporciona la IDE Netbeans podemos verificar el correcto funcionamiento (Figura 48).

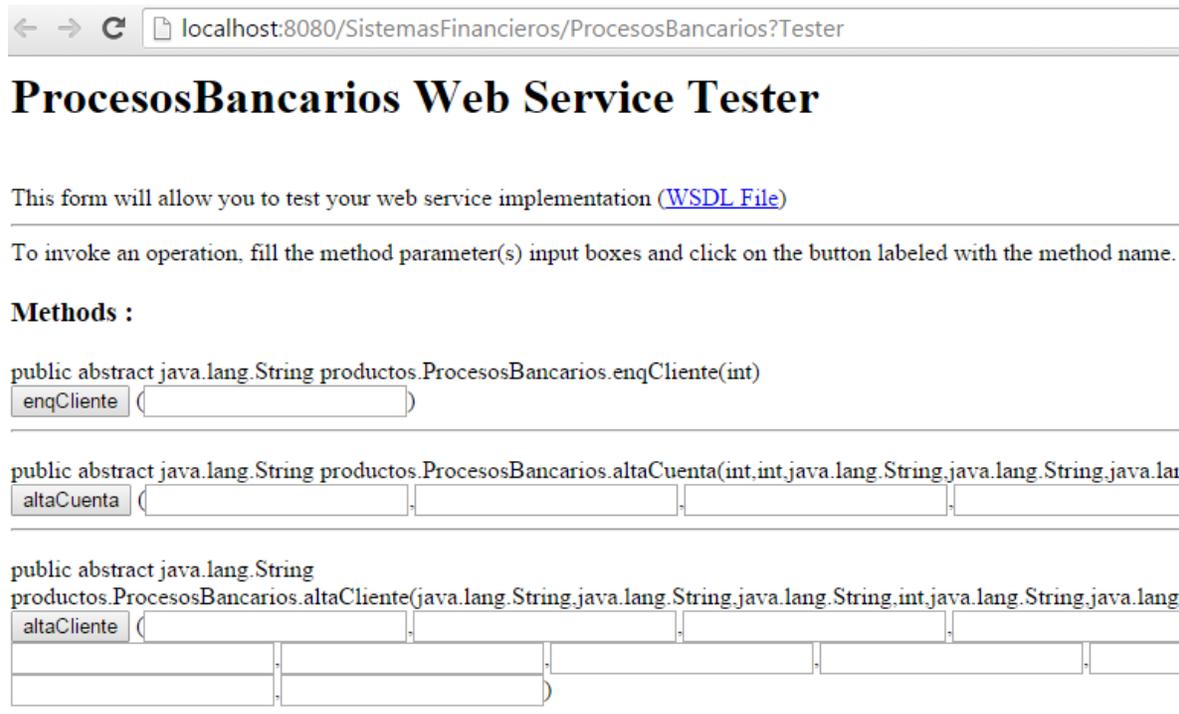


Figura 48 Verificación de Web Service

Pruebas Integrales:

Caso de prueba 1: Ingreso correcto del ejecutivo de cuenta a la sucursal

Esperado:

- Se podrá ingresar al sistema con un ejecutivo existente.
- No se podrá ingresar al sistema con ejecutivo no existente.

Prueba integral 1: Error de conexión (Figura 49).

Cuando el servidor por alguna razón no se encuentra disponible, la interfaz genera el siguiente aviso.

El intento de conexión no fue correcto. Inténtelo de nuevo.

Información de cuenta

Nombre de usuario:
yolitzin

Contraseña:

Mantenerme conectado

Iniciar sesión

Figura 49 Ingreso de ejecutivo

Prueba integral 2: Se requiere el usuario y contraseña para poder ingresar al sistema (Figura 50).

En los casos de contar con las credenciales correctas el sistema indica que los datos ingresados no son los correctos y por consecuencia no permite continuar.

MICROFINANCIERA

Quiénes somos Menú principal Acerca de

INICIAR SESIÓN

Especifique su nombre de usuario y contraseña. [Registrarse](#) si no tiene una cuenta.

- El nombre de usuario es obligatorio.
- La contraseña es obligatoria.

Información de cuenta

Nombre de usuario:
*

Contraseña:
*

Mantenerme conectado

Iniciar sesión

Figura 50 Prueba Ingreso de ejecutivo 2

Caso de prueba 2: Verificación de campos obligatorios (Figura 51).

MICROFINANCIERA [Iniciar sesión]

Quiénes somos Menú Principal Acerca de

ALTA DE CLIENTE

Datos del banco

No. de sucursal

Fecha de ingreso

*Fecha ingreso requerido

Datos generales del cliente

Nombre <input type="text"/>	Apellido paterno <input type="text"/>	Apellido materno <input type="text"/>
*Nombre cliente requerido	*Apellido paterno requerido	*Apellido materno requerido
Edad <input type="text"/>	Sexo <input type="text"/>	Estado civil <input type="text"/>
*Edad requerido		
Nacionalidad <input type="text"/>	RFC <input type="text"/>	CURP <input type="text"/>
	*RFC requerido	*CURP requerido
Tipo de identificación <input type="text"/>	Número de identificación <input type="text"/>	Fecha de nacimiento <input type="text"/>
*Tipo identificación requerido	*Número identificación requerido	*Fecha de Nac. requerido

Dirección del cliente

Calle <input type="text"/>	No. interior <input type="text"/>	No. exterior <input type="text"/>
Código postal <input type="text"/>	Delegación <input type="text"/>	Municipio <input type="text"/>
Estado <input type="text"/>	País <input type="text"/>	No. celular <input type="text"/>
Comprobante de domicilio <input type="text"/>	*País requerido	Correo electrónico <input type="text"/>
*Comprobante requerido		
No. comprobante de domicilio <input type="text"/>		
*No. comprobante requerido		

Guardar

Figura 51 Ingreso de cliente

Verificación de ingreso de cuentas con campos incompletos (Figura 52).

MICROFINANCIERA [Iniciar sesión]

Quiénes somos Menú Principal Acerca de

ALTA DE CUENTA

Número de sucursal 0000100001	Número de cliente 0000100001	Tipo de cuenta 1
Nombre CARMEN	Apellido paterno MENDEZ	Apellido materno MENDEZ
Comisión de apertura <input type="text"/>	Fecha apertura <input type="text"/>	Moneda <input type="text"/>
Ejecutivo de cuenta <input type="text"/>	*Fecha ingreso requerido	*Moneda requerido

Beneficiario

Nombre <input type="text"/>	Apellido paterno <input type="text"/>	Apellido materno <input type="text"/>
Parentesco <input type="text"/>	% de relación <input type="text"/>	

Perfil de cuenta

Depósito en efectivo <input type="text"/>	Depósito otros <input type="text"/>
Retiro en efectivo <input type="text"/>	Pago de cheques <input type="text"/>
Transferencias emitidas/recibidas <input type="text"/>	

Guardar

Figura 52 Ingreso de cuenta

Verificación de errores para campos incompletos de un crédito (Figura 53).

The screenshot shows a web interface for a microfinance institution. At the top, there is a blue header with the logo 'MICROFINANCIERA' and navigation links: 'Quiénes somos', 'Menú principal', and 'Acerca de'. Below the header is a Twitter icon. The main content area is titled 'ALTA DE CRÉDITO' and contains a form with the following fields and values:

Número de cliente	Número de cuenta	Envío documento (Si/No)
1000010003	1000010003	SI
Nombre	Apellido paterno	Apellido materno
CARMEN	LOPEZ	LOPEZ
Tipo de producto	Moneda	Tipo de tasa
NUEVO	MXN	FIJA
Plazo (Meses)	Monto	Tasa de interés
24	45000	
Línea permitida(Si/No)	Frecuencia de pago	
SI		

Below the form, there are two red error messages: '*Tasa interés requerido' and '*Frecuencia requerido'. At the bottom of the form is a 'Guardar' button.

Figura 53 Ingreso de crédito

Mediante las pruebas realizadas se garantiza que se cumple con los requerimientos solicitados por parte del cliente y se asegura también el funcionamiento de la Microfinanciera, ya que uno de los objetivos de esta etapa es descubrir los defectos de los componentes habiendo probado los programas individualmente y en conjunto.

Las pruebas realizadas abarcaron funcionales y no funcionales para verificar que el sistema realice lo que se le solicitó y cómo es que lo hace.

Las pruebas de caja negra mostradas son no funcionales en donde se fuerza a mostrar los errores posibles a una entrada errónea y correcta.

3.5 Funcionamiento y mantenimiento

Para el ingreso correcto de un crédito, se debe realizar la captura del cliente, cuenta y crédito en el orden correspondiente.

Captura de cliente (Figura 54).



ALTA DE CLIENTE

Datos del banco

No. de sucursal
10001

Fecha de ingreso
20151001



Datos generales del cliente

Nombre CARMEN	Apellido paterno MENDEZ	Apellido materno MENDEZ
Edad 32	Sexo FEMENINO	Estado civil SOLTERO
Nacionalidad MEXICANA	RFC CAMEM880311BP6	CURP CAMEM880311MMCMMLO2
Tipo de identificación IFE	Número de identificación 23213123123	Fecha de nacimiento 19880311

Dirección del cliente

Calle 1 DE MAYO	No. interior 7	No. exterior 17
Código postal 56604	Delegación SAN MIGUEL JACALONES	Municipio CHALCO
Estado MEXICO	País MEXICO	No. celular 5525624614
Comprobante de domicilio TELMEX		Correo electrónico car_2015@gmail.com
No. comprobante de domicilio 123213499898	<input type="button" value="Guardar"/>	

Figura 54 Ingreso de cliente

De ser necesario realizar actualizaciones a los datos del cliente, es posible hacerlo en la opción *Consulta de datos del cliente* (Figura 55).



CONSULTA DE CLIENTE

Datos del banco

No. de sucursal
10000010001

No. de cliente
10000010001





CONSULTA DE CLIENTE

Datos del banco

No. de sucursal
10000010001
No. de cliente
10000010001
Fecha de ingreso
20151010



Datos generales del cliente

Nombre CARMEN	Apellido paterno MENDEZ	Apellido materno MENDEZ
Edad 28	Sexo FEMENINO	Estado civil SOLTERA
Nacionalidad MEXICANA	RFC CAMEM880311BP6	CURP CAMEM880311MML02MMC
Tipo de identificación IFE	Número de identificación 1231239898	Fecha de nacimiento 19880311

Dirección del Cliente

Calle 1 DE MAYO	No. interior 7	No. exterior 17
Código postal 56604	Delegación CHALCO	Municipio CHALCO
Estado MEXICO	País MEXICO	No. celular 5525624614
Comprobante de domicilio TELMEX		Correo electrónico caro_2015@gmail.com
No. Comprobante de domicilio 213132112		

Guardar

Figura 55 Consulta de datos de cliente

Del mismo modo, existe la opción de eliminar los registros de los clientes que ya no forman parte de la *Microfinanciera* (Figura 56).



ELIMINAR REGISTROS

Cuentas Clientes Créditos

ID Registro 1000010001

Eliminar

Figura 56 Eliminar un cliente

Captura de cuenta (Figura 57).

MICROFINANCIERA

Quiénes somos Menú principal Acerca de

ALTA DE CUENTA

Número de sucursal 10101112	Número de cliente 1000100012	Tipo de cuenta NOMINA
Nombre CARMEN	Apellido paterno LOPEZ	Apellido materno LOPEZ
Comisión de apertura 5	Fecha apertura 20151010	Moneda MXN
Ejecutivo de cuenta ANDRE MALDONADO		
Beneficiario		
Nombre NA	Apellido paterno NA	Apellido materno NA
Parentezco NA	% de relación NA	
Perfil de cuenta		
Depósito en efectivo 10000	Depósito otros 100000	
Retiro en efectivo 100000	Pago de cheques 10000	
Transferencias Emitidas/ Recibidas 19000		

Guardar

Figura 57 Captura cuenta

Captura de crédito (Figura 58)

MICROFINANCIERA

Quiénes somos Menú principal Acerca de

ALTA DE CRÉDITO

Número de cliente 1000010003	Número de cuenta 1000010003	Envío documento (Si/No) SI
Nombre CARMEN	Apellido paterno LOPEZ	Apellido materno LOPEZ
Tipo de producto NUEVO	Moneda MXN	Tipo de tasa FIJA
Plazo (Meses) 24	Monto 24000	Tasa de interés 14.3
Línea permitida(Si/No) SI	Frecuencia de pago MENSUAL	

Guardar

Figura 58 Captura crédito

El desarrollo del proyecto de la *Microfinanciera* siguió de manera estricta la metodología adoptada lo cual garantiza el orden y la estructura a seguir para satisfacer las necesidades planteadas por el usuario, y asimismo garantiza la entrega en el tiempo pactada y de esta manera evitar desplazamientos de tiempos y personal para salir a producción.

Del mismo modo, la metodología nos ayuda a garantizar que el resultado esperado es aquel planteado por el usuario, ya que durante el proceso del análisis se presenta al usuario para conocer el alcance y las dimensiones son las esperadas y de esa manera realizar un correcto levantamiento de datos; en esta etapa es importante saber que el líder del proyecto deberá de realizar de manera correcta su función ya que de esto dependerá el resto del proyecto.

Dentro del diseño se evalúa la manera más sencilla de solucionar el requerimiento con el menor número de impacto en tiempo, costos y costos hombre, para que el desarrollo se realice de la manera más eficiente posible.

Las pruebas realizadas dejan como evidencia que el resultado es el esperado por el cliente, ya que cumple con los objetivos planteados.



Capítulo 4
Resultados, impacto y
conclusiones



4.1 Resultados

Durante la exposición del presente trabajo, se ha revisado el desarrollo de un sistema que es capaz de mejorar la calidad de los procesos de la *Microfinanciera* así como de optimizarlos con la finalidad de ofrecer un mejor servicio y más eficiente.

En este capítulo, se detalla el cumplimiento de los requerimientos planteados en la etapa inicial del desarrollo, así como el impacto que se ha producido en la actividad de la empresa.

Con la implementación del sistema se logran mejoras al proceso de asignación de créditos de la *Microfinanciera*, así como al seguimiento que se le da a los mismos mientras estén activos.

Podemos observar que se cumple con aquellos puntos que aseguran su funcionamiento en gran escala, es decir, cuando la *Microfinanciera* logró su objetivo de tener más sucursales fuera del estado en donde surge; como lo son escalabilidad, y adaptividad detalles que se cumplen al tener como base un sistema diseñado con un Web Service.

Los beneficios resultantes, son listados a continuación:

Minimizar errores de captura por parte de los usuarios así como evitar omisiones de datos obligatorios

En los formularios de entrada de la interfaz (alta de cliente, de cuenta y de crédito) se incluyen validaciones para orientar a los usuarios acerca del tipo de dato que se espera en cada campo, con esto se reduce el riesgo de escribir información no correspondiente o errónea en cada uno de ellos. Por ejemplo, en la Figura 59, pueden verse los mensajes de ayuda que envía el sistema para indicar que hay datos que deben ser capturados de manera obligatoria.

Igualmente, las interfaces incluyen controles que impiden ingresar información a la base de datos si el registro no cuenta con la información mínima necesaria para ello; así, por ejemplo, no pueden darse de alta clientes que no tengan RFC o CURP o que no estén capturados correctamente, etcétera.

Así pues, mediante el uso de validaciones se garantiza la correcta captura de los datos requeridos para ingresar un nuevo cliente, cuenta o crédito.

The screenshot shows a web interface for a microfinance institution. At the top, there is a blue header with the logo 'MICROFINANCIERA' and navigation links: 'Quiénes somos', 'Menú Principal', and 'Acercas de'. Below the header is a Twitter icon. The main content area is titled 'ALTA DE CLIENTE' and contains several sections of input fields:

- Datos del banco:** 'No. de sucursal' (100000100001) and 'Fecha de ingreso' (20151010).
- Datos generales del cliente:** 'Nombre' (CARMEN), 'Apellido paterno' (empty), 'Apellido materno' (MENDEZ), 'Edad' (27), 'Sexo' (FEMENINO), 'Estado civil' (SOLTERO), 'Nacionalidad' (MEXICANA), 'CURP' (CAMEM19880311MML02MC), 'Tipo de identificación' (IFE), 'RFC' (CAMEM19880311BP6), 'Número de identificación' (1232143214), and 'Fecha de nacimiento' (19880311).
- Dirección del cliente:** 'Calle' (1 DE MAYO), 'Código postal' (56604), 'Estado' (MEXICO), 'Comprobante de domicilio' (TELMEX), 'No. comprobante de domicilio' (1231241241), 'No. interior' (7), 'Delegación' (CHALCO), 'País' (MEXICO), 'No. exterior' (17), 'Municipio' (CHALCO), 'No. celular' (5525624614), and 'Correo electrónico' (caro2015@gmail.com).

A 'Guardar' button is located at the bottom center of the form. A red validation message '*Apellido paterno requerido' is displayed next to the 'Apellido paterno' field.

Figura 59 Validación de campos obligatorios

Erradicar la falta de objetividad por parte de los ejecutivos en cuanto a la asignación de créditos

Gracias a que los documentos entregados a la *Microfinanciera* por parte de los clientes son cargados digitalmente al sistema, se facilita el análisis de la información cuando esto sea necesario y, del mismo modo, se favorece que las decisiones de otorgamiento de créditos estén basadas en la información contenida en el sistema y que no se realice a juicio de los ejecutivos; esto permite tener una asignación más justa del financiamiento.

Además, gracias a que los documentos son cargados a una base de datos, éstos se asocian inequívocamente a cada cliente evitando así errores de integridad.

Como se muestra en la Figura 60, para evitar la falta de objetividad de los ejecutivos al momento de la asignación de un préstamo se han agregado validaciones para que en caso de contar con todos los datos y el saldo menor al máximo, éste se autorice de manera automática.

MICROFINANCIERA [Iniciar sesión]

Quiénes somos Menú Principal Acerca de

ALTA DE CRÉDITO

El crédito requiere de validación: Monto mayor al permitido

Número de cliente: 10000001010
Número de cuenta: 10000001010
Envió documento (Si/No): Si

MICROFINANCIERA [Iniciar sesión]

Quiénes somos Menú Principal Acerca de

ALTA DE CRÉDITO

El crédito fue autorizado

Número de cliente: 1000010002
Número de cuenta: 1000010002
Envió documento (Si/No): si

Figura 60 Validaciones automáticas para aceptar autorizar un crédito

Agilizar el proceso de aceptación de un crédito

Gracias a que la documentación, así como los datos de los clientes están cargados en un sistema de información, es posible tener un acceso más rápido a la información debido a la reducción de los tiempos de búsqueda. Así, cuando un registro ha sido capturado de manera correcta, pasa inmediatamente a la base de datos (Figura 61).

En ello radica la importancia del uso de una base de datos pues ésta permite indizar los registros contenidos en ella; así, se facilitan y agilizan las búsquedas con criterios específicos dentro de sus tablas. Se enfatiza, del mismo modo, la importancia que tiene el correcto diseño de la base, pues de esto dependerá la velocidad y veracidad de dichas búsquedas.

Número de cliente	Número de cuenta	Envió documento (Sí/No)
1000010002	1000010002	si
Nombre	Apellido paterno	Apellido materno
MARIO	DAVILA	DAVILA

Figura 61 Los registros autorizados se guardan en la base de datos

En casos extraordinarios, los créditos que no fueron autorizados deberán mostrar el motivo por el cual han sido rechazados

Ya sea porque no entreguen la documentación solicitada, porque no cumplan con los requisitos necesarios para poder ser candidatos a recibir un financiamiento, o por cualquier otra razón, en algunas ocasiones habrá créditos que no sean aprobados.

Sea cual sea la causa, si alguna solicitud es rechazada, los ejecutivos tienen acceso al registro (por un tiempo determinado) para ver el motivo por el cual no fue otorgado el préstamo.

Ello significa que los registros contenidos en la base de datos no son eliminados en el momento justo en que se vuelven no candidatos para recibir el apoyo de la empresa, sino que el nivel de los permisos de acceso a la información cambia. Es decir, el detalle de la información de dichos registros se vuelve visible únicamente para los usuarios (ejecutivos y/o gerentes) que tengan los permisos necesarios para verlos y así poder conocer el motivo del rechazo de la solicitud del cliente en cuestión (Figura 62).

The screenshot shows a web application header for 'MICROFINANCIERA' with navigation links: 'Quiénes somos', 'Menú Principal', and 'Acerca de'. Below the header is a Twitter icon and the title 'ALTA DE CRÉDITO'. A red message states: 'El crédito requiere de validación: Monto mayor al permitido'. Below this are six input fields arranged in two rows and three columns:

Número de cliente	Número de cuenta	Envió documento (Si/No)
10000001010	10000001010	SI
Nombre	Apellido paterno	Apellido materno
CARMEN	MENDEZ	MENDEZ

Figura 62 Motivos de rechazo de un crédito

Garantizar el mantenimiento constante de los registros en el sistema

Se estima que la *Microfinanciera* crezca en cuanto a número de sucursales y por ende en número de clientes; como es evidente, las bases de datos crecerán en su debida proporción por lo que la información contenida en el sistema debe mantenerse siempre actualizada y debidamente respaldada para evitar pérdidas o extracción de información errónea.

Con esto se entiende que la infraestructura de la *Microfinanciera* (tales como servidores), deben tener la capacidad suficiente para soportar el crecimiento del banco de datos.

La base de datos puede ser más grande y en caso de llegar a una cifra mayor a lo admisible, se podrá migrar a un servidor con mayor potencial (Figura 63).

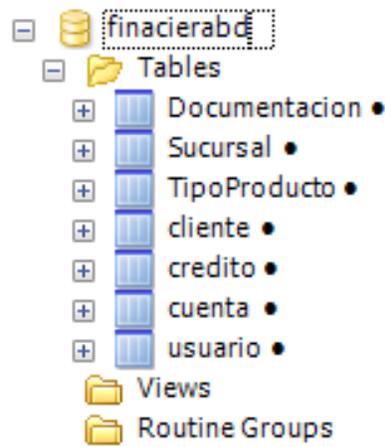


Figura 63 Estructura de la base de datos

Mediante el Web Service pueden crearse nuevos métodos de acuerdo a los nuevos requerimientos del cliente (Figura 64) sin necesidad de realizar un nuevo sistema.

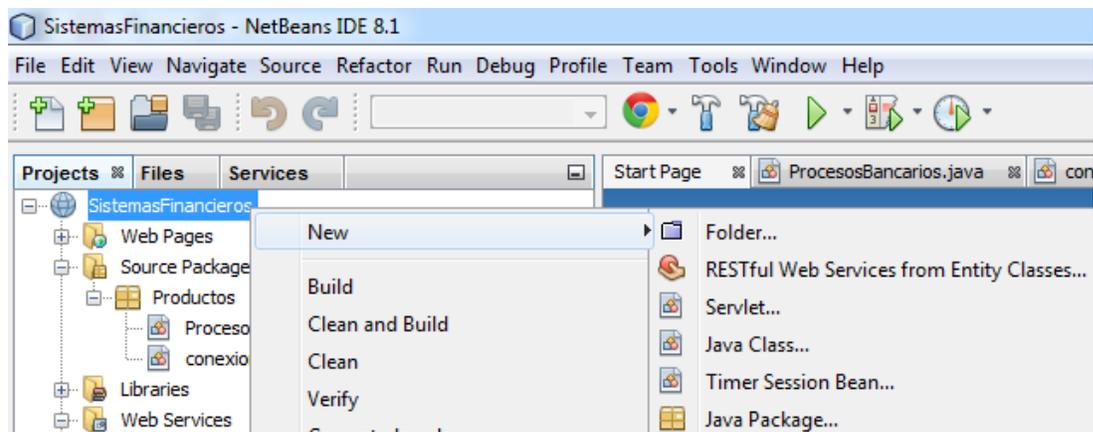


Figura 64 Estructura del Web Service

Garantizar la escalabilidad del sistema

Se espera que la *Microfinanciera* tenga un aumento en el número de sucursales, ello implica el crecimiento del número de usuarios que deben interactuar con el sistema (realizando consultas, ingresos y/o modificación de información), por lo que se ha preparado al sistema para que soporte el aumento del flujo de datos y la concurrencia.

Del mismo modo, con el crecimiento del sistema es importante garantizar la integridad de la información contenida en las bases de datos, algunas medidas para ello son evitando que más de un usuario pueda acceder a un mismo registro a la vez o manteniendo un control adecuado de los perfiles de cada usuario que hace uso del sistema (Figura 65Figura 65).

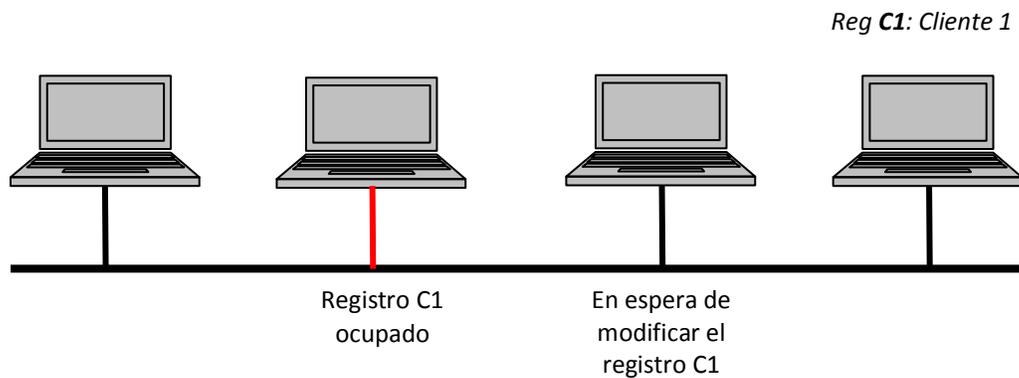


Figura 65 Integridad de la información y concurrencia

Garantizar la usabilidad de la interfaz hacia el sistema

El diseño de la interfaz se ha realizado de tal forma que sea amigable con el usuario y al mismo tiempo intuitivo, es decir, que le permita adaptarse rápidamente al manejo y funcionalidad del sistema.

Así, de acuerdo con lo anterior, si existen errores de captura de información por parte de los usuarios, el sistema les envía mensajes de ayuda para que les sirva de apoyo y puedan percatarse de qué les hace falta o qué están capturando erróneamente, evitando con ello errores posteriores en la autorización del crédito e incongruencia de datos. En caso contrario, y si el(los) crédito(s) no requieren de revisiones especiales, quedan autorizados automáticamente por el sistema (Figura 66).

The figure displays two screenshots of a web application interface for credit authorization. Both screenshots feature a dark blue header with the text 'MICROFINANCIERA' and a navigation menu with 'Quiénes somos', 'Menú Principal', and 'Acerca de'. A '[Iniciar sesión]' link is visible in the top right corner of the first screenshot. Below the header, a Twitter icon is present. The main content area is titled 'ALTA DE CRÉDITO' and includes the heading 'El crédito fue autorizado'. The form contains six input fields arranged in two rows and three columns. The first row includes 'Número de cliente', 'Número de cuenta', and 'Envío documento (Si/No)'. The second row includes 'Nombre', 'Apellido paterno', and 'Apellido materno'. In the first screenshot, the values are 1000010002, 1000010002, si, MARIO, DAVILA, and DAVILA. In the second screenshot, the values are 1000010003, 1000010003, si, MARIO, JIMENEZ, and DAVILA. A small house icon is located to the right of the 'Envío documento' field in the first screenshot.

Figura 66 En caso de no haber errores de captura, el registro queda autorizado

Verificación de ingreso por Web Service

Información: datos: 1000010003, 1000010003, MARIO, JIMENEZ, DAVILA, 1, MXN, 15500, 24,

si, 12, SI

Información: Se conecto con la base jdbc:mysql://localhost:3306/finacierabd

Información: datos: insert into credito

values('878189632','1000010002','1000010002','MARIO','JIMENEZ','DAVILA','1','MXN','15500','24',
'si','12','Fija','SI')

Información: datos: 1000010002, 1000010002, MARIO, DAVILA, DAVILA, 1, MXN, 15500, 24, si,
12, SI

Información: Se conecto con la base jdbc:mysql://localhost:3306/finacierabd

Información: datos: insert into credito

values('878189632','1000010002','1000010002','MARIO','JIMENEZ','DAVILA','1','MXN','15500','24',
'si','12','Fija','SI')

Garantizar la integridad de la información utilizada

Como es bien sabido, no todos los usuarios tienen el mismo nivel de acceso a los datos sino que poseen ciertos niveles de permiso. El uso de esta jerarquización asegura la protección del sistema pues únicamente permite realizar modificaciones a los datos a un número específico y conocido de usuarios.

Del mismo modo, es fundamental resaltar la importancia del adecuado diseño de las bases de datos pues debe existir una relación inequívoca de los registros de cada una de las tablas para evitar errores en el sistema.

Ejemplo: Mediante una consulta podemos asegurarnos que los datos ingresados a la base de datos son los correctos (Figura 67).

MICROFINANCIERA

Quiénes somos | Menú Principal | Acerca de



CONSULTA DE CUENTA

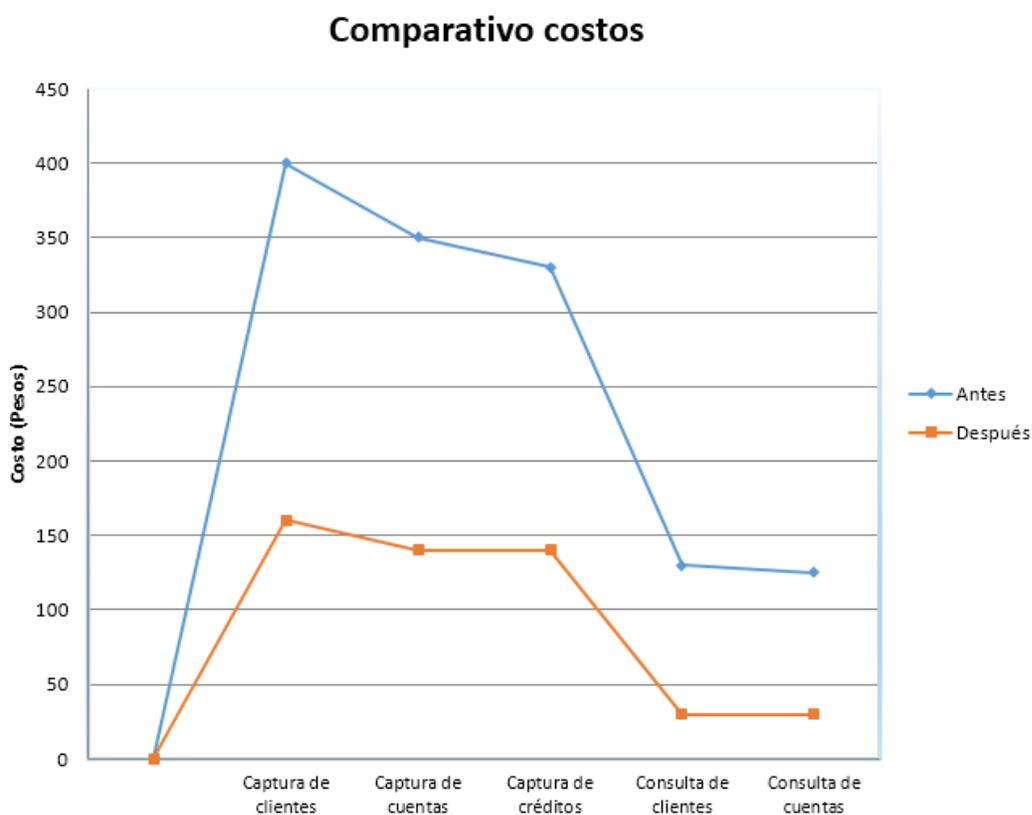
Número de cuenta	<input type="text" value="1000010001"/>	Número de cliente	<input type="text" value="1000010001"/>	Tipo de cuenta	<input type="text" value="1"/>
Número de sucursal	<input type="text" value="1000010001"/>	Apellido paterno	<input type="text" value="JIMENEZ"/>	Apellido materno	<input type="text" value="JIMENEZ"/>
Nombre	<input type="text" value="MARIO"/>	Fecha apertura	<input type="text" value="20151010"/>	Moneda	<input type="text" value="MXN"/>
Comisión de apertura	<input type="text" value="NA"/>				
Ejecutivo de cuenta	<input type="text" value="101101"/>				
Beneficiario					
Nombre	<input type="text" value="SAUL"/>	Apellido paterno	<input type="text" value="JIMENEZ"/>	Apellido materno	<input type="text" value="JIMENEZ"/>
Parentesco	<input type="text" value="HIJO"/>	% de relación	<input type="text" value="100"/>		
Perfil de cuenta					
Depósito en efectivo	<input type="text" value="SI"/>	Depósito otros	<input type="text" value="SI"/>		
Retiro en efectivo	<input type="text" value="SI"/>	Pago de cheques	<input type="text" value="SI"/>		
Transferencias emitidas/recibidas	<input type="text" value="SI"/>				

Figura 67 Consulta de registros

Minimizar costos en la captura de información así como de compartir información con diferentes sucursales

Se reduce el costo de archivado gracias a que no es necesario el almacenamiento físico de los documentos solicitados a los clientes (ya no es requerido guardar copias de los documentos); por otro lado, el almacenamiento virtual favorece la indexación de la información lo que ayuda a reducir los tiempos de búsqueda de datos específicos.

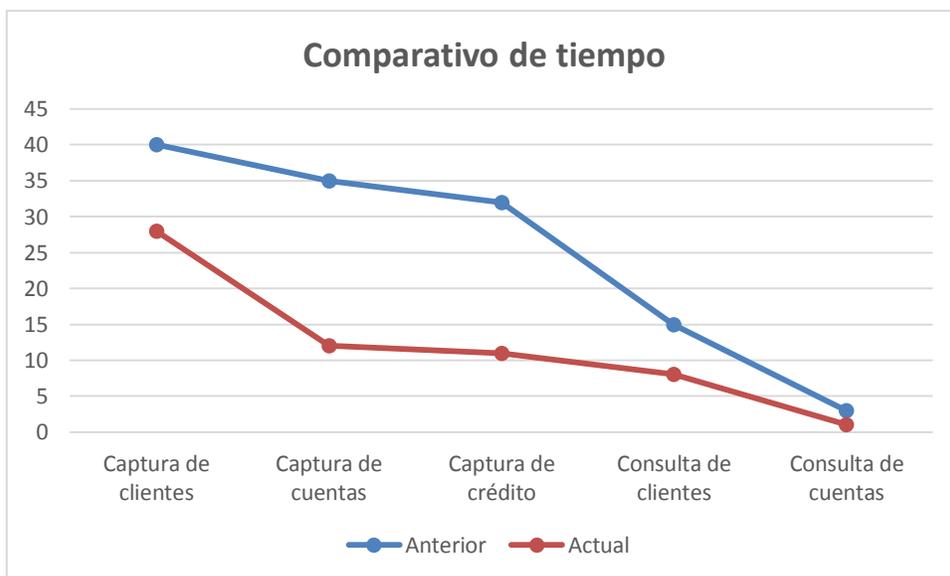
Además, la información puede ser consultada desde cualquier sucursal siempre y cuando tenga los permisos adecuados para ello.



Minimizar tiempo en los procesos de alta de cliente y crédito

Al ser el sistema capaz de realizar validaciones en los datos que están siendo capturados por los ejecutivos, se reduce la probabilidad de que existan errores en la información que se está quedando guardada en las bases de datos de la *Microfinanciera*.

Con ello, se está evitando también realizar correcciones manuales cuando el proceso de asignación de montos ya se encuentra avanzado.



Garantizar la disponibilidad del sistema en cualquiera de las sucursales de la Microfinanciera

En horarios de servicio de la *Microfinanciera* (cualquiera que éste sea), el sistema debe estar siempre disponible para los usuarios, permitiendo así darle continuidad al negocio. Además, el sistema debe ser capaz de soportar la concurrencia de los usuarios.

Ejemplo: Se ingresa dos créditos simultáneamente en diferentes sucursales (Figura 68).

Sucursal 1:



MICROFINANCIERA

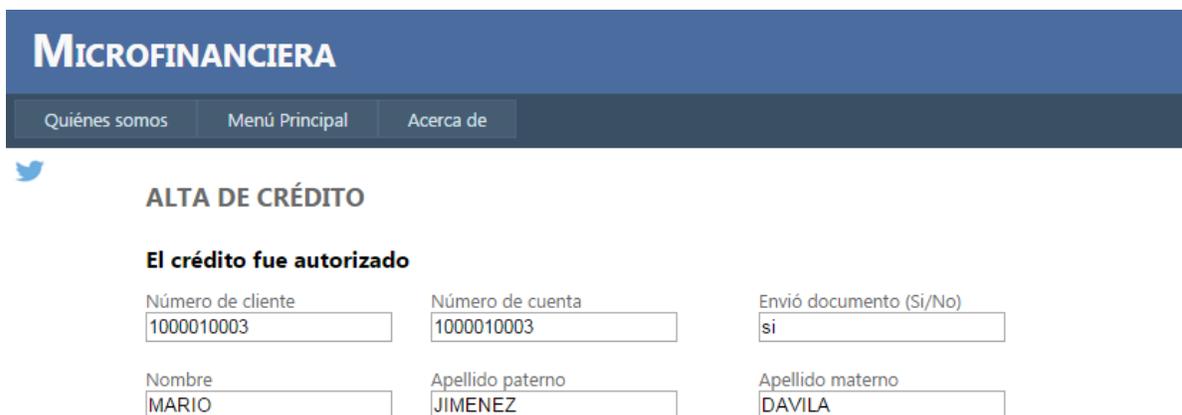
Quiénes somos | Menú Principal | Acerca de

ALTA DE CRÉDITO

El crédito fue autorizado

Número de cliente	Número de cuenta	Envió documento (Si/No)
1000010002	1000010002	si
Nombre	Apellido paterno	Apellido materno
MARIO	DAVILA	DAVILA

Sucursal 2:



MICROFINANCIERA

Quiénes somos | Menú Principal | Acerca de

ALTA DE CRÉDITO

El crédito fue autorizado

Número de cliente	Número de cuenta	Envió documento (Si/No)
1000010003	1000010003	si
Nombre	Apellido paterno	Apellido materno
MARIO	JIMENEZ	DAVILA

Figura 68 Captura simultánea de registros

Se logra observar como ingresan ambas solicitudes al Web Service:

Verificación de ingreso por Web Service

Información: datos: 1000010003, 1000010003, MARIO, JIMENEZ, DAVILA, 1, MXN, 15500, 24, si, 12, SI

```
Información: Se conecto con la base jdbc:mysql://localhost:3306/finacierabd
Información: datos: insert into credito
values('878189632','1000010002','1000010002','MARIO','JIMENEZ','DAVILA','1','MXN','15500','24',
'si','12','FIJA','SI')
Información: datos: 1000010002, 1000010002, MARIO, DAVILA, DAVILA, 1, MXN, 15500, 24, si,
12, SI
Información: Se conecto con la base jdbc:mysql://localhost:3306/finacierabd
Información: datos: insert into credito
values('878189632','1000010002','1000010002','MARIO','JIMENEZ','DAVILA','1','MXN','15500','24',
'si','12','FIJA','SI')
```

4.2 Impacto

Antes de la implementación de un sistema, los flujos en los procesos de la empresa, desde el alta del cliente hasta la liquidación del crédito, se llevaba a cabo manualmente. Esto provocaba, ya sea por descuido o por falta de comunicación entre los empleados, algunos errores que afectaban de alguna forma la continuidad de los procedimientos del negocio.

Igualmente, el archivado de información era ineficiente debido a que se tenía que invertir un tiempo considerablemente largo para la búsqueda de datos o para adicionar documentos al registro de un cliente ya existente.

Ahora, con el uso del sistema, una de las principales ventajas que se representan en la *Microfinanciera*, es el ahorro de recursos dentro de ella, esto se logra reduciendo el tiempo necesario para dar de alta un crédito (desde el alta del cliente hasta la autorización y asignación de un crédito) y utilizando medios de almacenamiento virtuales en lugar de físicos.

Este tipo de almacenamiento permite, además, acceder de manera más ágil a los datos y reducir el tiempo necesario para realizar búsquedas de información de los clientes tanto en la sucursal dueña de los mismos (aquella en donde fueron dados de alta) como en alguna otra sucursal que requiera algún detalle específico (siempre y cuando el (los) ejecutivo(s) tenga(n) los permisos suficientes para acceder a ella).

Por otro lado, el uso del sistema permite minimizar posibles fallas de los empleados de la *Microfinanciera* tales como falta de captura de datos relevantes, así como la existencia de errores de transcripción de información, además, se verifica que esta captura sea congruente al validar, por ejemplo, que el RFC coincida con los datos del nombre del cliente y fecha de nacimiento (gracias a las validaciones que realiza al momento de introducir datos), con lo que se garantiza la integridad de la información capturada.

Una utilidad adicional de tener estas validaciones automáticas (es decir, que no hay necesidad de la “objetividad” de un analista de créditos) es que se elimina casi por completo el hecho de que las decisiones del otorgamiento de créditos sean tomadas a juicio de los ejecutivos. Adicional a esto, es importante destacar que, al ser automáticas las autorizaciones de los créditos (excepto en casos excepcionales en que el monto del crédito sea mayor al comúnmente otorgado), también se evita la falta de objetividad.

Sin embargo, debido a que los empleados no están familiarizados con la nueva plataforma para realizar sus procesos diarios, se optó por elegir una interfaz que fuera lo suficientemente amigable con ellos para facilitar y acelerar el aprendizaje de la misma.

Dado lo anterior, se desarrolló una vista en .NET que tiene un entorno gráfico que permite una interacción sencilla con el sistema.

Por otro lado, se les asignarán distintos niveles de permisos a los usuarios (dependiendo de sus funciones y de las necesidades de la *Microfinanciera*) para proteger la información personal y sensible de los posibles clientes (es decir, aquellos que apenas van a buscar un financiamiento) y los clientes ya existentes (es decir, aquellos que tienen un crédito activo) en la *Microfinanciera*.

Así, por ejemplo, un ejecutivo podrá acceder únicamente a la aplicación para dar de alta clientes, cuentas y créditos. En casos excepcionales, tendrá la posibilidad de realizar la aceptación de un crédito.

En cambio, un gerente o analista de créditos podrá acceder a la aplicación para autorizar algún crédito (en caso de ser necesario), así como también tiene permisos para realizar consultas a la información de los clientes.

Finalmente, es importante resaltar que los costos serán minimizados debido a que no se necesitará fotocopiar cada uno de los documentos proporcionados por el cliente sino únicamente realizar un escaneo y un almacenamiento en la base de datos, lo cual garantizará, además, la accesibilidad a ellos sin importar la hora o el lugar donde se realice la consulta.

Además de evitar el gasto de papelería, se evita el gasto de mensajería ya que no es necesario el envío de la información cada vez que se requiera, de esta manera se logra un cliente satisfecho con la *Microfinanciera* ya que se reducen tiempos de respuesta.

4.3 Conclusiones

Una microfinanciera, en general, trabaja con datos sensibles de las personas a quienes les está prestando un servicio por lo que la seguridad de los documentos que son entregados deben tener una importancia primordial.

Otro aspecto al que se le debe prestar especial atención es la eficiencia de sus procesos, pues de ellos depende que los clientes no presenten quejas por los servicios que están recibiendo, lo que, inherentemente implica un aumento en el número de clientes.

Así pues, el software diseñado garantiza a la *Microfinanciera* que será capaz de ofrecer un mejor servicio a sus clientes y que les podrá dar soporte en momentos complicados; así como también le ayudará en el crecimiento del negocio.

Además, la versatilidad del diseño y del desarrollo garantiza que en el futuro esta *Microfinanciera* pueda crecer sin necesidad de realizar un cambio drástico a su sistema debido a que es escalable y de fácil implementación.

Del mismo modo, al hacer uso de una tecnología como el Web Service, se garantiza que puedan ser implementadas diferentes interfaces de usuario (montadas en distintas plataformas) sin necesidad de realizar cambios en los niveles más bajos del sistema.

Siendo así, se puede apreciar que una de las ventajas que se logran con este desarrollo es el aumento de la capacidad de captura de datos, lo que nos garantiza un mejor servicio y un mayor crecimiento tanto en la cantidad de clientes como en el número de sucursales de la *Microfinanciera*.

Se puede apreciar también que la reducción de tiempos se ve reflejada directamente en los costos y en la calidad del servicio ofrecido, con lo cual puede concluirse que al hacer uso de las tecnologías informáticas vigentes, mejoran en gran medida los servicios y las ganancias de la empresa.

Cabe destacar que haciendo uso de la metodología del modelo de ciclo de vida clásico, se ha logrado tanto una mejor planeación como una temprana determinación de costos del proyecto, esto gracias a que se tienen que analizar todos los elementos que se necesitan para el desarrollo y los tiempos respectivos para llevar a cabo cada uno de ellos.

En la planeación, se garantiza al cliente que cada componente está siendo considerado; en el diseño, se modela la solución de cada requerimiento y el desarrollo se lleva a cabo siguiendo las pautas que nos marca dicho modelo, teniendo una importancia relevante las

pruebas y el proceso de validación pues de ellos depende que el sistema sea verdaderamente de calidad.

Por último, se puede decir que las bases cimentadas durante el tiempo que se cursó en la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, han ayudado a tener un crecimiento social y técnico, además de que fomentaron un pensamiento lógico que permitirá nuestro desarrollo profesional, creando sistemas que benefician a la sociedad en general.

ANEXO A

Índice de Figuras

Figura 1. Alta de un crédito en la <i>Microfinanciera</i>	- 3 -
Figura 2. Control actual del proceso de captura de datos	- 4 -
Figura 3. Aceptación o rechazo de créditos a juicio de los ejecutivos	- 4 -
Figura 4. Organigrama de una microfinanciera	- 19 -
Figura 5. Promoción del crédito	- 22 -
Figura 6. Consulta en sociedades de información crediticia	- 23 -
Figura 7. Ciclo del crédito	- 26 -
Figura 8. Actividades fundamentales para el desarrollo de software	- 27 -
Figura 9. Modelo en cascada modificado.....	- 29 -
Figura 10. Flujo de los componentes del lenguaje de programación Java.....	- 37 -
Figura 11. Logo JAVA.....	- 39 -
Figura 12. Evolución de los lenguajes de programación	- 40 -
Figura 13. Encapsulamiento	- 42 -
Figura 14. Características de lenguaje de programación JAVA	- 44 -
Figura 15. Importancia de Web Service	- 46 -
Figura 16. Arquitectura del protocolo SOAP	- 49 -
Figura 17. Componentes del Framework .NET	- 51 -
Figura 18. .NET Framework en contexto	- 53 -
Figura 19. Ejecución CLR	- 54 -
Figura 20. Principales Clases de las librerías NET Framework	- 55 -
Figura 21. IDE Visual Studio.....	- 56 -
Figura 22. Bases de datos y su importancia	- 58 -
Figura 23. Historia de las bases de datos	- 60 -
Figura 24 Plan de trabajo	- 67 -
Figura 25 Diagrama de Gantt.....	- 68 -
Figura 26. Diseño arquitectónico.....	- 70 -
Figura 27. Diagrama Entidad-Relación.....	- 73 -
Figura 28 Web Service para <i>Microfinanciera</i>	- 74 -
Figura 29 Conexión a base de datos	- 75 -
Figura 30 Sistema de la <i>microfinanciera</i>	- 75 -
Figura 31 Flujo de aceptación de crédito	- 76 -
Figura 32 Interfaz de usuario.....	- 77 -
Figura 33 Tablas por diagramas.....	- 79 -
Figura 34 Explorador del proyecto.....	- 81 -
Figura 35 Métodos de servicios.....	- 82 -
Figura 36 Menú principal de la interfaz	- 83 -
Figura 37 Opción de Quiénes Somos	- 84 -

Figura 38 Validación de campos para el alta de cliente	- 85 -
Figura 39 Captura de cliente	- 86 -
Figura 40 Pantalla de alta de cuenta	- 87 -
Figura 41 Alta de crédito	- 87 -
Figura 42 Consulta de cliente	- 88 -
Figura 43 Consulta de cuenta	- 88 -
Figura 44 Consulta de crédito.....	- 89 -
Figura 45 Eliminación de registros.....	- 89 -
Figura 46 Pantalla “Acerca de”	- 90 -
Figura 47 Interfaz de alta de crédito.....	- 91 -
Figura 48 Verificación de Web Service.....	- 92 -
Figura 49 Ingreso de ejecutivo	- 93 -
Figura 50 Prueba Ingreso de ejecutivo 2.....	- 93 -
Figura 51 Ingreso de cliente	- 94 -
Figura 52 Ingreso de cuenta	- 94 -
Figura 53 Ingreso de crédito.....	- 95 -
Figura 54 Ingreso de cliente	- 96 -
Figura 55 Consulta de datos de cliente	- 97 -
Figura 56 Eliminar un cliente	- 97 -
Figura 57 Captura cuenta	- 98 -
Figura 58 Captura crédito.....	- 98 -
Figura 59 Validación de campos obligatorios.....	- 102 -
Figura 60 Validaciones automáticas para aceptar autorizar un crédito	- 103 -
Figura 61 Los registros autorizados se guardan en la base de datos.....	- 104 -
Figura 62 Motivos de rechazo de un crédito.....	- 105 -
Figura 63 Estructura de la base de datos	- 106 -
Figura 64 Estructura del Web Service	- 106 -
Figura 65 Integridad de la información y concurrencia	- 107 -
Figura 66 En caso de no haber errores de captura, el registro queda autorizado	- 108 -
Figura 67 Consulta de registros	- 110 -
Figura 68 Captura simultánea de registros	- 113 -

ANEXO B

Índice de tablas

Tabla 1. Accesibilidad geográfica y demográfica por niveles de ingreso (medianas).....	- 11 -
Tabla 2. Uso efectivo y cuentas según nivel de ingreso.....	- 12 -
Tabla 3. Las finanzas populares en México: formal e informal	- 13 -
Tabla 4. Divisiones del sistema financiero	- 15 -
Tabla 5. Estructura organizacional de una microfinanciera	- 18 -
Tabla 6. Comparativa de grupos solidarios y crédito individual.....	- 20 -
Tabla 7. Capacidades de una arquitectura de referencia SOA	- 48 -

ANEXO C

Referencias

- 1.- INEGI, MCS-ENIGH 2010, con base en la metodología del CONEVAL sobre el cálculo multidimensional de la pobreza en México 2010.
- 2.- Pagina web: INEGI <http://cuentame.inegi.org.mx/poblacion/habitantes.aspx?tema=P>
- 3.- Según informa el Consejo Nacional de Evaluación de la Política de Desarrollo Social (Coneval) en el anexo estadístico 2012 de la pobreza en México.
<http://www.coneval.gob.mx/Medicion/MP/Paginas/Pobreza-urbana-en-M%C3%A9xico-.aspx>
- 4.- Decreto con Fuerza de Ley Especial de Asociaciones Cooperativas (Gaceta Oficial N° 37.285 del 18-09-2001.)
- 5.- Datos Banco Mundial, Beck, Kunt y Martínez Perla M.S. (<http://www.bancomundial.org/>)
- 6.- <http://economics.mit.edu/files/530>
- 7.- Datos Banco Mundial, Beck, Kunt y Martínez Perla M.S. (<http://www.bancomundial.org/>)
- 8.- <http://www.condusef.gob.mx/index.php/otros-servicios-y-productos/945.html>
- 9.- Mansell Carstens, Catherine. *Las finanzas populares en México. El redescubrimiento de un sistema financiero olvidado*. CEMLA. Milenio, ITAM. México, 1995
- 10.- De acuerdo a lo publicado por BANXICO en:
<http://www.banxico.org.mx/divulgacion/sistema-financiero/sistema-financiero.html>
- 11.- De acuerdo a la CONDUSEF
<http://www.condusef.gob.mx/index.php/instituciones-financieras/otros-sectores/uniones-de-credito>
- 12.- De acuerdo a la CONDUSEF
<http://www.condusef.gob.mx/index.php/instituciones-financieras/otros-sectores/sociedades-financieras-de-objeto-limitado-sofoles>
- 13.- De acuerdo a la CONDUSEF,
www.condusef.gob.mx/index.php/instituciones-financieras/sociedades-financieras-de-objeto-multiple
- 14.- De acuerdo a la CONDUSEF
www.condusef.gob.mx/index.php/instituciones-financieras/64-sociedades-financieras-de-objeto-multiple/633-caracteristicas-de-una-sofom
- 15.- De acuerdo a la CONDUSEF
<http://www.condusef.gob.mx/index.php/instituciones-financieras/sociedades-cooperativas-de-ahorro-y-credito-popular/916-sociedades-cooperativas-de-ahorro-y-prestamo-scap>
- 16.- De acuerdo a la CNBV
<http://www.cnbv.gob.mx/SECTORES-SUPERVISADOS/SECTOR-POPULAR/Preguntas-Frecuentes/Paginas/Sociedades-Financieras-Populares.aspx>
- 17.- <http://sofinco.com.mx/images/Figuras.pdf>
- 18.- Ceballos, Francisco Javier. *Java 2 Curso de programación*. Alfaomega, 4a edición. México 2011.
- 19.- http://www.cad.com.mx/historia_del_lenguaje_java.htm
- 20.- <http://docs.oracle.com/javaee/6/tutorial/doc/gijti.html>
- 21.- <http://img.redusers.com/imagenes/libros/lpcu143>
- 22.- [http://msdn.microsoft.com/es-es/library/zw4w595w\(v=vs.110\).aspx](http://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.110).aspx)
- 23.- <http://img.redusers.com/imagenes/libros/lpcu143/capitulogratis.pdf>
- 24.- Pressman S. Roger. *Ingeniería de software. Un enfoque práctico*. McGrawHill, 5th edición, México 2002
- 25.- <http://dicyg.fi-c.unam.mx:8080/lalo/aed-teo/FasesDS.pdf>
- 26.- <http://www.enterate.unam.mx/Articulos/2006/febrero/arquitect.htm>
- 27.- WebCab Components, About Component Based Development
- 28.- <http://webcabcomponents.com/componentization.shtml>
- 29.- Schneiderman, Ben; Plaisant, Catherine. *Designing the user interface*. Pearson, 5a edición.

ANEXO D

Índice de Acrónimos y Siglas

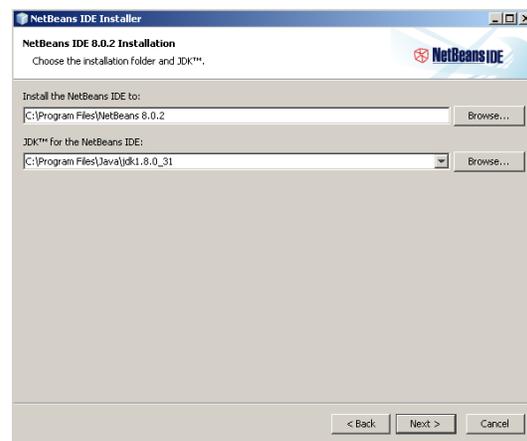
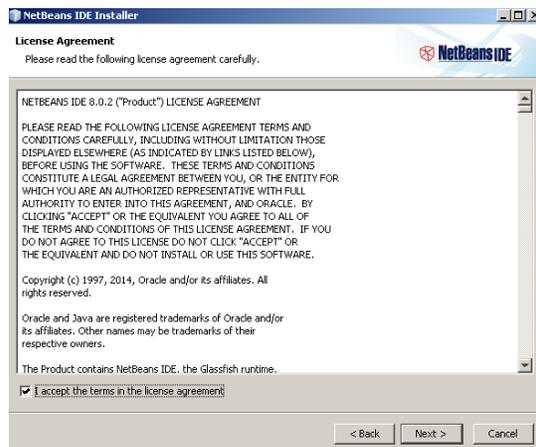
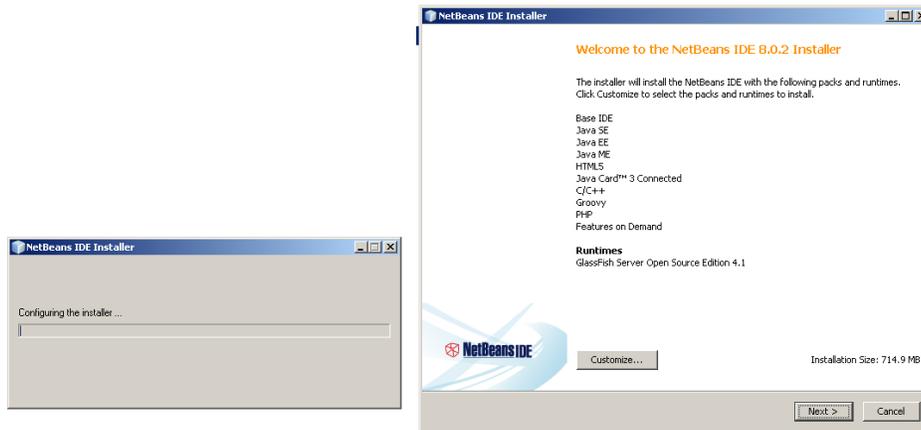
BANSEFI	Banco del Ahorro Nacional y Servicios Financieros
BCL	Biblioteca de Clases Base
CLI	Common Language Infrastructure
CNBV	Comisión Nacional Bancaria y de Valores
CONDUSEF	Comisión Nacional para la Protección y Defensa de los Usuarios y Servicios Financieros
CTS	Common Type System, Sistema de Tipos Común
DBMS	Database Management System
DLL	Dynamic-Link Library
EXE	Executable
GUI	Graphical User Interface
IDE	Integrated Development Environment
IIS	Internet Information Services
JIT	Just-In-Time
POO	Programación Orientada a Objetos
SAR	Cuentas de ahorro para el retiro (Sistema de Ahorro para el Retiro)
SCAP	Sociedades Cooperativas de Ahorro y Préstamo
SHCP	Secretaría de Hacienda y Crédito Público
SOAP	Simple Object Access Protocol
SOFINCO	Sociedades Financieras Comunitarias
SOFIPO	Sociedades Financieras Populares
SOFOL	Sociedades Financieras de Objeto Limitado
SOFOM	Sociedades Financieras de Objeto Múltiple
UDIS	Unidades de inversión
WF	Workflow Foundation
WSDL	Web Service Description Language
XML	eXtensible Markup Language

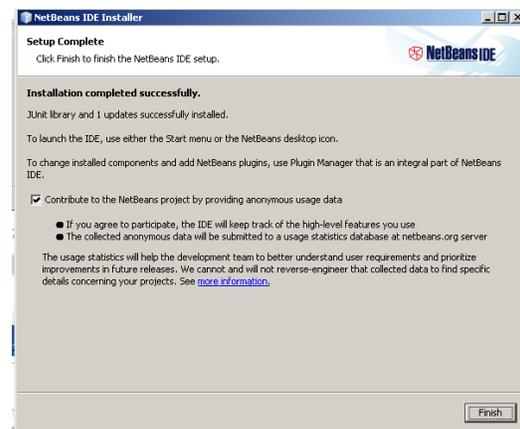
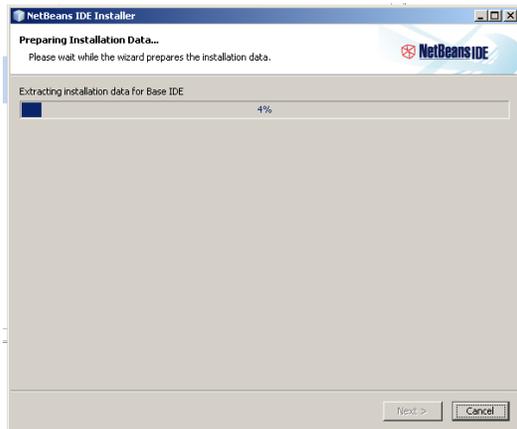
ANEXO E

Instalación de componentes

Instalación de IDE de JAVA. Para este caso es NETBEANS 8.0.2

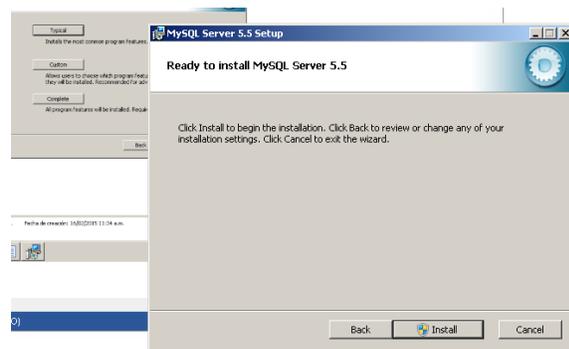
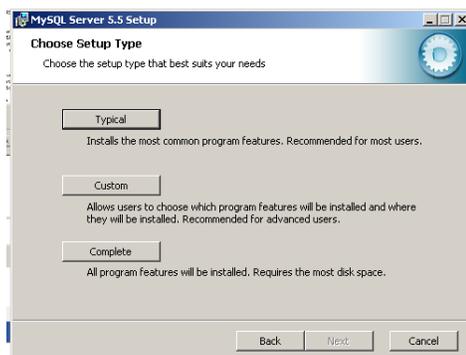
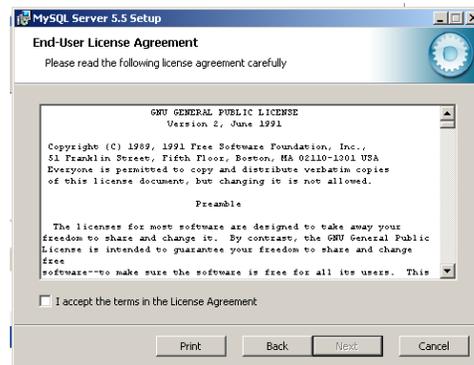
Para la instalación de Netbeans se deberá dar seguimiento a cada opción disponible.

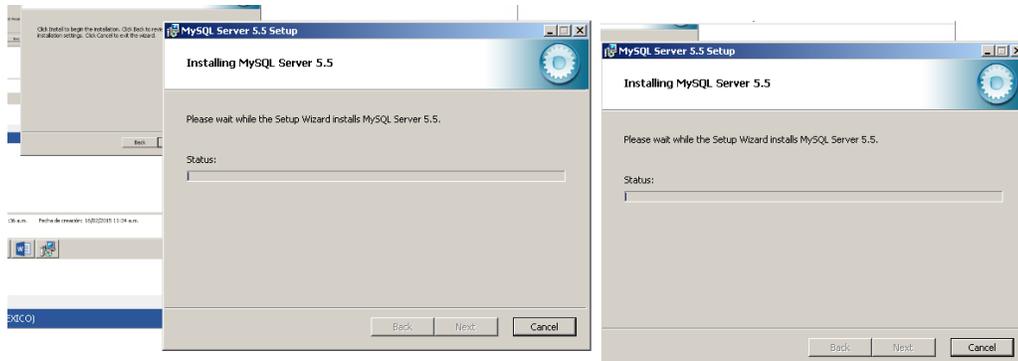




Instalación de SQL (MySQL en este caso).

Seguir los siguientes pasos para lograr una correcta instalación de MySQL.





Instalación de MySQL:

